Multicriteria Decision Aid and Artificial Intelligence

VISIT…

LANZAROTE Caliente.com

Multicriteria Decision Aid and Artificial Intelligence

Links, Theory and Applications

Edited by

Michael Doumpos and Evangelos Grigoroudis

Technical University of Crete, Greece

WILEY

A John Wiley & Sons, Ltd., Publication

This edition first published 2013

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Doumpos, Michael.
Multicriteria decision aid and artificial intelligence : links, theory and applications / Michael Doumpos, Evangelos Grigoroudis.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-119-97639-4 (hardback)
1. Multiple criteria decision making. 2. Artificial intelligence. I. Grigoroudis, Evangelos. II. Title.
T57.95.D578 2013
658.4’033–dc23

2012040171

A catalogue record for this book is available from the British Library.

ISBN: 978-1-119-97639-4

Set in 10/12pt Times by Laserwords Private Limited, Chennai, India

Contents

Preface

In the rapidly evolving technological and business environment, decision making becomes increasingly more complex from many perspectives. For instance, environmental and sustainable development issues have risen but the related policies, priorities, goals, and socio-economic tradeoffs are not well defined or understood in depth. Furthermore, technological advances provide new capabilities in many areas such as telecommunications, web-based technologies, transportation and logistics, manufacturing, energy management, and worldwide trade. Finally, the economic turmoil increases the uncertainties in the global business environment, and has direct impact on all socio-economic and technological policies.

Such a challenging environment calls for the implementation of enhanced tools, processes, and techniques for decision analysis and support. Clearly, these should take into account the aforementioned multiple diverse aspects, in combination with the specific priorities and goals set by decision and policy makers. This has always been a prerequisite for providing decision support in a realistic context. But it is not enough anymore, as decision support technologies should nowadays also accommodate a variety of new (often crucial) requirements, such as distributed decision making, the handling of massive and increasingly complex data and structures, as well as the computational difficulties that arise in building and using models and systems, which are realistic enough to represent the dynamic nature of existing problems and challenges posed by new ones.

Multicriteria decision aid (MCDA) has evolved significantly over the past decades as a major discipline in operations research, dealing with providing decision support in complex, ill-structured problems involving multiple (conflicting) criteria. MCDA is involved in all aspects of the decision process, including problem structuring, model building, formulation of recommendations, implementation and support. Issues like preference modeling, the construction of proper sets of criteria and their measurement, the characterization of different criteria aggregation models, the development of effective interactive solution techniques, and the implementation of sophisticated methods in user-friendly decision support systems, have traditionally been at the core of MCDA research.

Among the many exciting trends developing in the area of MCDA, the one focused on exploring the connections of MCDA with other disciplines, is particularly interesting for providing integrated decision support in the complex context described above. In this framework, artificial intelligence (AI) has attracted much interest. Nowadays, AI is a broad field within which one can identify several major research areas, including

among others, machine learning/data mining, soft computing, evolutionary computation, knowledge engineering and management, expert systems, symbolic reasoning, cognitive systems, etc. Even though, AI research is mostly focused on predictive modeling and the development of technological systems that imitate human behavior, the methods and techniques developed in this field have much to offer towards meeting the new requirement for decision support described above. This potential has been acknowledged by researchers working in MCDA and AI, and has led to a growing trend involved with a unification of ideas developed (often) independently in the two fields. Among others, the results of this unification trend can be identified in the development of new decision modeling forms and paradigms, advanced solution techniques for complex decision problems, new approaches for preference elicitation and learning, as well as implementations in integrated intelligent systems.

Nevertheless, despite the vast amount of research published on MCDA and AI, the research on their integration is scattered across different sources, which are often oriented toward different readerships from the one or the other field. Having that in mind, the aim set for the preparation of this edited volume, was to present in a unified manner the various capabilities that the integration of MCDA and AI provide, from different decision support perspectives, focusing on state-of-the-art research advances in this area, the comprehensive coverage of the existing literature, as well as applications in several fields. The book includes 14 chapters organized into five parts, covering all these topics in a comprehensive and rigorous manner.

The first two chapters provide detailed overviews of the use of AI methods in MCDA. In particular, the first chapter by Doumpos and Zopounidis, is focused on the computational intelligence paradigm. The chapter begins with an introduction to the basic concepts and principles of MCDA and then proceeds with an up-to-date review on the uses of computational intelligence approaches in MCDA. The review focuses on the methodological contributions of statistical learning, fuzzy systems, and evolutionary computation, in areas such as preference modeling, preference disaggregation analysis, multiobjective optimization, and decision making under uncertainty.

In the second chapter, Phillips-Wren extends the overview covering AI techniques and approaches in the context of decision support systems. The chapter first introduces the fundamentals of human decision making, followed by a presentation of the decision support systems philosophy, as computer systems that utilize data, models, and knowledge to solve complex decision problems. Then, the contribution of intelligent technologies is discussed covering areas such as neural networks, fuzzy logic, evolutionary computing, expert systems, and intelligent agents. The chapter closes with the introduction of a framework for evaluating the success of intelligent decision support systems in a multicriteria context.

The following two chapters cover AI techniques and technologies, which are particularly useful in the context of preference modeling as well as for the development of decision support systems. In particular, the chapter by Comes, Wijngaards, and Schultmann, involves multicriteria decision support systems, in a distributed setting. The authors focus on strategic decision making problems, which are characterized by high complexity and uncertainty. Examples of such decision situations are discussed and the key challenges are identified. To meet these challenges, the authors propose a framework combining techniques from MCDA with scenario-based reasoning. The framework is illustrated through a decision making situation involving emergency management.

The next chapter, by Valls, Moreno, and Borràs, is involved with the representation and management of the user preferences in decision support systems. The chapter illustrates how a semantic-based approach can be implemented to store and exploit the personal preferences of a user in complex domains. This approach is based on ontologies, which enable the representation of the domain's elements in a machine understandable manner. The authors analyze over 30 semantic-based recommender systems and review different ontology-based models for preference representation, as well as algorithms for learning the user profile. The chapter also discusses the way MCDM techniques can be combined with ontology-based user profiles in recommended systems.

Chapters 5–7 present the contributions of popular AI paradigms in constructing new types of decision aiding models. In Chapter 5, Hanne focuses on neural networks. Neural networks have been one of the most widely used and successful AI techniques. The chapter presents the main concepts and types of neural networks and reviews their use in various aspects of MCDA.

Chapter 6, by Szelag, Greco, and Słowiński, is devoted to rule-based models. The authors focus on ranking problems, where the objective is to rank a set of alternatives from the best to the worst ones. Models expressed in the form of decision rule are widely used in data mining and machine learning for classification. This chapter illustrates how rough set theory, a popular machine learning technique, can be extended to multicriteria ranking problems. The proposed methodology is based on the dominance-based rough set approach and it enables the development of decision rule models from decision examples.

In Chapter 7, Boujelben and De Smet analyze the applications of evidence theory in MCDA. Evidence theory has primarily been developed in the context of AI as a generalization of subjective probability theory, which is particularly suitable for problems under uncertainty and total ignorance. Thus, evidence theory provides a convenient framework for modeling and combining imperfect information. The chapter describes five multicriteria methods based on evidence theory and presents new concepts that have been developed within this modeling approach with specific applications in MCDA.

The following three chapters of the book are devoted to computational intelligence methods for multiobjective optimization. This part of the book starts with Chapter 8, by López Jaimes and Coello Coello, which is devoted to evolutionary algorithms for multiobjective optimization. This is one of the most active research topics in operations research in general and MCDA in particular. López Jaimes and Coello Coello focus on interactive procedures, in which the decision-maker has an active role in the solution process. The chapter presents a categorized review of recent multiobjective evolutionary algorithms designed to work as interactive optimization methods.

In the following chapter, Yun and Nakayama illustrate how data envelopment analysis (DEA) can be used in the context of multiobjective optimization combined with computational intelligence techniques. DEA is a popular approach for analyzing the efficiency of decision making units. Its underlying philosophy is closely related to multiobjective optimization and Pareto optimality. The authors introduce a generalized data envelopment analysis (GDEA) model and present several methods for combining GDEA and computational intelligence techniques for generating approximate Pareto optimal solutions. The methodology also enables the identification of the most interesting part of Pareto optimal solutions, as well as the improvement of the operation of popular computational optimization techniques.

In Chapter 10, Sakawa presents a comprehensive overview of fuzzy multiobjective linear programming. The chapter introduces modeling formulations for problems where the decision-maker has fuzzy goals, as well as cases where the parameters in the description of the objective functions and the constraints are fuzzy. Interactive solution techniques are also presented for these classes of problems. The chapter also discusses stochastic multiobjective linear programming problems and illustrates how they can be transformed into deterministic ones using a probability maximization model together with chance constrained conditions.

The last four chapters of the book are application-oriented, illustrating how the combination of AI and MCDA techniques contribute in addressing complex real-world decision making problems from various fields. In Chapter 11, Delias and Matsatsinis present a multicriteria methodology for supporting resource sharing in virtual organizations. The problem is considered in a cloud computing context, where specific computing resources should be managed in order to meet the needs of the clients. The proposed methodology adopts a multi-agent system design approach, with the ultimate goal of the methodology being the modeling of the collective preferences of the agents (clients). The methodology is illustrated through an example application to a data center that seeks to optimize the application environments that it hosts.

The following chapter, by Sarı, Öztayşi, and Kahraman, presents the combination of fuzzy sets with MCDA techniques for the evaluation of warehouse location sites. Fuzzy set theory enables the formal modeling of uncertainty, vagueness, and imprecision that characterizes many ill-structured complex decision problems. The authors discuss the applicability of fuzzy sets in several MCDA techniques, and illustrate how a modeling approach combining type-2 fuzzy sets with the analytic hierarchy process can be employed to a warehouse selection problem.

In Chapter 13, Diakaki and Grigoroudis illustrate the use of genetic algorithms in the optimization of energy efficiency in building in a multiobjective context. The authors review the existing literature on the use of such optimization techniques in improving the energy efficiency of buildings and present an application case study regarding the minimization of the initial investment cost and the increase of the energy savings in a building.

The book closes with a chapter by Vassiliadis and Dounias on the use of a computational intelligence multiobjective optimization approach for portfolio optimization. In a traditional portfolio optimization setting an investor seeks to construct a portfolio of assets that maximizes return for a given level of risk, which leads to a bi-objective quadratic optimization model. The authors consider this problem using the case where constraints are imposed on the number of assets in the portfolio and examine the use of new risk measures. In order to construct the set of efficient portfolios, a genetic algorithm is employed combined with a local search procedure. Computational results are presented using a data set involving stocks from the New York stock exchange.

Sincere thanks must be expressed to all the authors who have devoted considerable time and effort to prepare excellent comprehensive works of high scientific quality and value. Without their help it would have been impossible to prepare this book in line with the high standards that we set from the very beginning of this project.

Michael Doumpos, Chania, Greece
Evangelos Grigoroudis, Chania, Greece

July 2012

Notes on Contributors

Mohamed Ayman Boujelben, GIAD, Faculté des Sciences Economiques et de Gestion de Sfax, Université de Sfax, Tunisia.

Joan Borràs, ITAKA (Intelligent Technologies for Advanced Knowledge Acquisition) Research Group, Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain, and Science & Technology Park for Tourism and Leisure, Vila-Seca, Catalonia, Spain.

Carlos A. Coello Coello, CINVESTAV-IPN, Departamento de Computación, Evolutionary Computation Group (EVOCINV), México D.F., Mexico.

Tina Comes, Institute for Industrial Production, Karlsruhe Institute of Technology, Germany.

Pavlos Delias, Department of Accountancy, Kavala Institute of Technology, Kavala, Greece.

Yves De Smet, CoDE-SMG, Ecole polytechnique de Bruxelles, Université libre de Bruxelles, Belgium.

Christina Diakaki, Department of Production Engineering and Management, Technical University of Crete, Chania, Greece.

Michael Doumpos, Department of Production Engineering and Management, Technical University of Crete, Chania, Greece.

Georgios Dounias, Management and Decision Engineering Laboratory, Department of Financial and Management Engineering, Business School, University of the Aegean, Greece.

Salvatore Greco, Department of Economics and Business, University of Catania, Italy.

Evangelos Grigoroudis, Department of Production Engineering and Management, Technical University of Crete, Chania, Greece.

Thomas Hanne, University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland.

Cengiz Kahraman, Department of Industrial Engineering, Istanbul Technical University, Turkey.

Antonio López Jaimes, CINVESTAV-IPN, Departamento de Computación, Evolutionary Computation Group (EVOCINV), México D.F., Mexico.

Nikolaos Matsatsinis, Department of Production Engineering and Management, Technical University of Crete, Chania, Greece.

Antonio Moreno, ITAKA (Intelligent Technologies for Advanced Knowledge Acquisition) Research Group, Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain.

Hirotaka Nakayama, Department of Intelligence and Informatics, Konan University, Kobe, Japan.

Başar Öztayşi, Department of Industrial Engineering, Istanbul Technical University, Turkey.

Gloria Phillips-Wren, Sellinger School of Business and Management, Loyola University Maryland, Baltimore, USA.

Masatoshi Sakawa, Department of System Cybernetics, Graduate School of Engineering, Hiroshima University, Japan.

Frank Schultmann, Institute for Industrial Production, Karlsruhe Institute of Technology, Germany.

Roman Słowiński, Institute of Computing Science, Poznań University of Technology, Poland, and Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland.

Marcin Szelag, Institute of Computing Science, Poznań University of Technology, Poland.

İrem Uçal Sarı, Department of Industrial Engineering, Istanbul Technical University, Turkey.

Aida Valls, ITAKA (Intelligent Technologies for Advanced Knowledge Acquisition) Research Group, Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Catalonia, Spain.

Vassilios Vassiliadis, Department of Financial and Management Engineering, Business School, University of the Aegean, Greece.

Niek Wijngaards, Thales Research & Technology Netherlands, D-CIS Lab, Delft, The Netherlands.

Yeboon Yun, Faculty of Environmental and Urban Engineering, Kansai University, Osaka, Japan.

Constantin Zopounidis, Department of Production Engineering and Management, Technical University of Crete, Chania, Greece.

Part I

THE CONTRIBUTIONS OF INTELLIGENT TECHNIQUES IN MULTICRITERIA DECISION AIDING

1

Computational intelligence techniques for multicriteria decision aiding: An overview

Michael Doumpos and Constantin Zopounidis
Department of Production Engineering and Management, Technical University of Crete, Greece

1.1 Introduction

Real world decision-making problems are usually too complex and ill-structured to be considered through the examination of a single criterion, attribute or point of view that will lead to an 'optimal' decision. In fact, such a single-dimensional approach is merely an oversimplification of the actual nature of the problem at hand, and it can lead to unrealistic decisions. A more appealing approach would be the simultaneous consideration of all pertinent factors that are related to the problem. However, through this approach some very essential issues/questions emerge: how can several and often conflicting factors be aggregated into a single evaluation model? Is this evaluation model unique and/or 'optimal'? In addressing such issues, one has to bear in mind that each decision-maker (DM) has his/her own preferences, experiences, and decision-making policy.

The field of multicriteria decision aid (MCDA) is devoted to the study of problems that fit the above context. Among others, MCDA focuses on the development and implementation of decision support tools and methodologies to confront complex decision problems involving multiple criteria, goals or objectives of conflicting nature. It has to

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

be emphasized through, that MCDA techniques and methodologies are not just some mathematical models aggregating criteria that enable one to make optimal decisions in an automatic manner. Instead, MCDA has a strong decision support focus. In this context the DM has an active role in the decision-modeling process, which is implemented interactively and iteratively until a satisfactory recommendation is obtained that fits the preferences and policy of a particular DM or a group of DMs.

Even though MCDA has developed as a major and well-distinguished field of operations research, its interaction with other disciplines has also received much attention. This is understood if one considers the wide range of issues related to the decision process, which the MCDA paradigm addresses. These involve among others the phases of problem structuring, preference modeling, the construction and characterization of different forms of criteria aggregation models, as well as the design of interactive solution and decision aid procedures and systems. The diverse nature of these topics often calls for an interdisciplinary approach.

A significant part of the research on the connections of MCDA with other disciplines has focused on intelligent systems. Over the past decades enormous progress has been made in the field of artificial intelligence, in areas such as expert systems, knowledge-based systems, case-based reasoning, fuzzy logic, and data mining. This chapter focuses on computational intelligence, which has emerged as a distinct sub-field of artificial intelligence involved with the study of adaptive mechanisms to enable intelligent behavior in complex and changing environments (Engelbrecht 2002). Typical computational intelligence paradigms include machine learning algorithms, evolutionary computation and nature-inspired computational methodologies, as well as fuzzy systems. We provide an overview of the main contributions of popular computational intelligence approaches in MCDA, covering areas such as multiobjective optimization, preference modeling, and model building through preference disaggregation.

The rest of the chapter is organized as follows: Section 1.2 presents an introduction to the MCDA paradigm, its main concepts and methodological streams. Section 1.3 is devoted to the overview of the connections between MCDA and computational intelligence, focusing on three main fields of computational intelligence, namely statistical learning/data mining, fuzzy set theory, and metaheuristics. Finally, Section 1.4 concludes the chapter and discusses some future research directions.

1.2 The MCDA paradigm

1.2.1 Modeling process

The major goal of MCDA is to provide a set of criteria aggregation methodologies that enable the development of decision support models considering the DM's preferential system and judgment policy. Achieving this goal requires the implementation of complex processes. Most commonly, these processes do not lead to optimal solutions/decisions, but to satisfactory ones that are in accordance with the DM's policy. Roy (1985) introduced a general framework that covers all aspects of the MCDA modeling philosophy (Figure 1.1).

The first level of the process, involves the specification of a set A of feasible alternative solutions for the decision problem at hand. This set can be continuous or discrete. In the

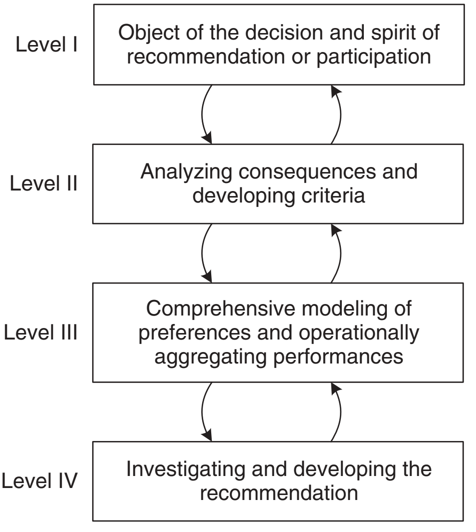

Figure 1.1 The MCDA modeling process.

former case, it is specified through a set of constraints. In the case where A is discrete, it is assumed that the DM can list the alternatives which will be subject to evaluation within the given decision-making framework. The form that the output of the analysis should have is also defined at the first phase of the process. This involves the selection of an appropriate decision 'problematic', which may involve: (a) the choice of the best alternative or a set of good alternatives; (b) the ranking of the alternatives from the best to the worst ones; (c) the classification of the alternatives into predefined categories; and (d) the description of the alternatives and their characteristics.

The second stage involves the identification of all factors related to the decision. MCDA assumes that these factors have the form of criteria. A criterion is a real function f measuring the performance of the alternatives on each of their individual characteristics. The set of selected criteria $\{f_1, \ldots, f_n\}$ must form a consistent family of criteria. A consistent family of criteria is characterized by the following properties (Bouyssou 1990):

- Monotonicity: If alternative $\mathbf{x}$ is preferred over alternative $\mathbf{y}$, the same should also hold for any alternative $\mathbf{z}$ such that $f_k(\mathbf{z}) \geq f_k(\mathbf{x})$ for all k.
- Completeness: If $f_k(\mathbf{x}) = f_k(\mathbf{y})$ for all criteria, then the DM should be indifferent between alternatives $\mathbf{x}$ and $\mathbf{y}$.
- Nonredundancy: The set of criteria satisfies the nonredundancy property if the elimination of any criterion results to the violation of monotonicity and/or completeness.

Once a consistent family of criteria has been specified, the next step is to proceed with the specification of the criteria aggregation model that meets the requirements of the problem. Finally, the last stage involves all the necessary supportive actions needed for the successful implementation of the results of the analysis and the justification of the model's recommendations.

1.2.2 Methodological approaches

MCDA provides a wide range of methodologies for addressing decision-making problems of different types. The differences between these methodologies involve the form of the models, the model development process, and their scope of application. On the basis of these characteristics, Pardalos *et al*. (1995) suggested the following four main streams in MCDA research:

- Multiobjective mathematical programming.
- Multiattribute utility/value theory.
- Outranking relations.
- Preference disaggregation analysis.

The following subsections provide a brief overview of these methodological streams.

1.2.2.1 Multiobjective mathematical programming

Multiobjective mathematical programming (MMP) extends the well-known single objective mathematical programming framework to problems involving multiple objectives. Formally, a MMP problem has the following form:

$$\begin{array}{ll} \max & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})\} \\ \text{subject to:} & \mathbf{x} \in \mathcal{A} \end{array} \tag{1.1}$$

where $\mathbf{x}$ is the vector of the decision variables, $f_1, f_2, \ldots, f_n$ are the objective functions (in maximization form), and $\mathcal{A}$ is the set of feasible solutions defined through multiple constraints.

In a MMP context, the objectives are assumed to be in conflict, which implies that it is not impossible to find a solution that maximizes all the objectives simultaneously. In that regard, efficient solutions (Pareto optimal or nondominated solutions) are of interest. A solution $\mathbf{x}^*$ is referred to as efficient if there is no other solution $\mathbf{x}$ that dominates $\mathbf{x}^*$, i.e., $f_k(\mathbf{x}) \geq f_k(\mathbf{x}^*)$ for all k and $f_j(\mathbf{x}) > f_j(\mathbf{x}^*)$ for at least one objective j. An overview of the MMP theory and different techniques for finding Pareto optimal solutions can be found in the books of Steuer (1985), Miettinen (1998), Ehrgott and Gandibleux (2002), and Ehrgott (2005).

An alternative approach to model multiobjective optimization problems is through goal programming formulations. In the context of goal programming a function of the deviations from some pre-specified goals is optimized. The goals are set by the DM and may represent ideal points on the objectives, some benchmark or reference points, or a set of satisfactory target levels on the objectives that should be met as closely as possible. The general form of a goal programming formulation is the following:

$$\begin{array}{lll} \min & \mathcal{F}(d_k^+, d_k^-; \mathbf{w}) & \\ \text{subject to:} & \mathbf{x} \in \mathcal{A} & \\ & f_k(\mathbf{x}) + d_k^+ - d_k^- = s_k, & k = 1, \ldots, n \\ & d_k^+, d_k^- \geq 0, & k = 1, \ldots, n \end{array} \tag{1.2}$$

where s_k is the target level (goal) set for objective k, d_k^+ and d_k^- are the deviations from the target, and $\mathcal{F}$ is a function of the deviations, which is parameterized by a vector $\mathbf{w}$ of

weighting coefficients. These coefficients may either represent the trade-offs between the deviations corresponding to different objectives or indicate a lexicographic ordering of the deviations' significance (pre-emptive goal programming). An overview of the theory and applications of goal programming can be found in Aouni and Kettani (2001), Jones and Tamiz (2002), as well as in the book of Jones and Tamiz (2010).

1.2.2.2 Multiattribute utility/value theory

Multiattribute utility/value theory (MAUT/MAVT) extends the traditional utility theory to the multidimensional case.[1] MAVT has been one of the cornerstones of the development of MCDA and its practical applications. The objective of MAVT is to model and represent the DM's preferential system into a value function $V(\mathbf{x})$, where $\mathbf{x}$ is the vector with the data available over a set of n evaluation criteria. The value function is defined on the criteria space, such that:

$$\begin{aligned} V(\mathbf{x}) > V(\mathbf{y}) &\Rightarrow \mathbf{x} \succ \mathbf{y} \\ V(\mathbf{x}) = V(\mathbf{y}) &\Rightarrow \mathbf{x} \sim \mathbf{y} \end{aligned} \tag{1.3}$$

where $\succ$ denotes preference and $\sim$ denotes indifference. The most commonly used form of value function is the additive one:

$$V(\mathbf{x}) = \sum_{k=1}^{n} w_k v_k(x_k) \tag{1.4}$$

where $w_k \geq 0$ is the trade-off constant for criterion k (usually the trade-off constants are assumed to sum up to one) and $v_k(x_k)$ is the corresponding marginal value function, which defines the partial value (performance score) of the alternatives on criterion k, in a predefined scale (e.g., in [0, 1]). If the marginal value function is assumed to be linear the additive model reduces to a simple weighted average of the criteria. Keeney and Raiffa (1993) present in detail the theoretical principles of MAVT under both certainty and uncertainty, and discuss the independence conditions that characterize different types of value models (e.g., additive, multiplicative, multi-linear).

1.2.2.3 Outranking techniques

The foundations of the outranking relation theory (ORT) have been set by Bernard Roy during the late 1960s through the development of the ELECTRE family of methods (ELimination Et Choix Traduisant la REalité; Roy 1968). Since then, ORT has been widely used by MCDA researchers, mainly in Europe. All ORT techniques operate in two major stages. The first stage involves the development of an outranking relation, whereas the second stage involves the exploitation of the outranking relation in order to perform the evaluation of the alternatives for choice, ranking, and classification purposes.

An outranking relation can be defined as a binary relation used to estimate the strength of the preference for an alternative $\mathbf{x}$ over an alternative $\mathbf{y}$. In comparison with MAVT, outranking techniques have two special features:

[1] The term 'utility' is used in the context of decision making under uncertainty, whereas the term 'value' is preferred for decisions in a certain environment. Henceforth, the term 'value' will be used throughout the chapter.

- An outranking relation is not necessarily transitive: in MAVT models the evaluation results are transitive. On the other hand, models developed on the basis of outranking relations allow intransitivities.
- An outranking relation is not complete: the main preference relations used in a MAVT modeling framework involve preference and indifference as defined in (1.3). In addition to these two relations, outranking methods also consider the incomparability relation, which arises when comparing alternatives with very special characteristics and diverse performance on the criteria.

The most popular methods implementing the outranking relations framework are the ELECTRE methods (Roy 1991), as well as the PROMETHEE methods (Brans and Mareschal 2005), with different variants for addressing choice, ranking and classification problems.

1.2.2.4 Preference disaggregation analysis

The development of the MCDA model can be performed through direct or indirect procedures. The former are based on structured communication sessions between the analyst and the DM, during which the analyst elicits specific information about the DM's preferences (e.g., weights, trade-offs, goals, etc.). The success of this approach is heavily based on the willingness of the DM to participate actively in the process, as well as the ability of the analyst to guide the interactive process in order to address the DM's cognitive limitations. This kind of approach is widely used in situations involving decisions of strategic character.

However, depending on the selected criteria aggregation model, a considerable amount of information may be needed by the DM. In 'repetitive' decisions, where time limitations exist, the above direct approach may not be applicable. Disaggregation methods (Jacquet-Lagrèze and Siskos 2001) are very helpful in this context. Disaggregation methods use regression-like techniques to infer a decision model from a set of decision examples on some reference alternatives, so that the model is as consistent as possible with the actual evaluation of the alternatives by the DM. This model inference approach provides a starting basis for the decision-aiding process. If the obtained model's parameters are in accordance with the actual preferential system of the DM, then the model can be directly applied to new decision instances. On the other hand, if the model is consistent with the sample decisions, but its parameters are inconsistent with the DM's preferential system (which may happen if, for example, the decision examples are inadequate), then the DM has a starting basis upon which he/she can provide recommendations to the analyst about the calibration of the model in the form of constraints about the parameters of the model. Thus, starting with a model that is consistent with a set of reference examples, an interactive model calibration process is invoked.

Jacquet-Lagrèze and Siskos (1982) introduced the paradigm of preference disaggregation in the context of decision aiding through the development of the UTA method (UTilité Additive), which enables the development of evaluation models in the form of an additive value function for ranking purposes. A comprehensive review of this

methodological approach of MCDA can be found in Jacquet-Lagrèze and Siskos (2001) and Siskos *et al*. (2005). Recent research has focused on extensions covering:

- other types of decision models, including among others outranking models (Doumpos and Zopounidis 2002b 2004 Mousseau *et al*. 2001), and rule-based models (Greco *et al*. 2001);
- other decision problematics (e.g., classification, Doumpos and Zopounidis 2002a);
- new modeling forms in the context of robustness decision-making (Dias *et al*. 2002),(Greco *et al*. 2008b).

1.3 Computational intelligence in MCDA

Computational intelligence has evolved rapidly over the past couple of decades and it is now considered as a distinct sub-field that emerged within the area of artificial intelligence. Duch (2007) discusses the unique features of computational intelligence as opposed to the artificial intelligence paradigm, analyzes the multiple aspects of computational intelligence and introduces a definition of the field as 'the science of solving non-algorithmizable problems using computers or specialized hardware.' Craenen and Eiben (2003) view artificial intelligence and computational intelligence as two complementary fields of 'machine intelligence.' In their view, artificial intelligence is mostly concerned with knowledge-based approaches whereas computational intelligence is a different stream involved with non-knowledge-based principles.

In the following subsections, we focus on three major computational intelligence paradigms, namely statistical learning/data mining, fuzzy sets, and metaheuristics, which all have been extremely popular among researchers and practitioners involved with the area of computational intelligence. We analyze the contributions of the paradigms within the context of decision-making problems by overviewing their connections with MCDA.

1.3.1 Statistical learning and data mining

Hand *et al*. (2001) define data mining as 'the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.' Statistical learning plays an important role in the data mining process, by describing the theory that underlies the identification of such relationships and providing the necessary algorithmic procedures.

Modern statistical learning and data mining adopt an *algorithmic modeling culture* as described by Breiman (2001), in which the focus is shifted from data models to the characteristics and predictive performance of learning algorithms. This approach is very different from the MCDA paradigm (a discussion of the similarities and differences in the context of the preference disaggregation approach of MCDA can be found in Doumpos and Zopounidis 2011b as well as in the work of Waegeman *et al*. 2009). Nevertheless, the algorithmic developments in statistical learning and data mining, such as the focus on the analysis of large scale data sets, as well as the wide range of different

types of generalized modeling forms employed in these fields, provide new capabilities in the context of MCDA.

1.3.1.1 Artificial neural networks

Artificial neural networks (ANNs) can be considered as directed acyclic graphs with nodes (neurons) organized into layers. The most popular feed-forward architecture consists of a layer of input nodes, a layer of output nodes, and a series of intermediate processing layers. The input nodes correspond to the information that is available for every input vector, whereas the output nodes provide the recommendations of the network. The nodes in the intermediate (hidden) layers are parallel processing units that define the input–output relationship. Every neuron at a given layer receives as input the weighted average of the outputs of the neurons at the preceding layer and maps it to an output signal through a predefined transformation function.

Depending on the topology of the network and the selection of the neurons' transformation functions, a neural network can model real functions of arbitrary complexity. This flexibility has made ANNs a very popular modeling approach in addressing complex real-world problems in engineering and management. This characteristic has important implications for MCDA, mainly with respect to modeling general preference structures.

Within this context, ANNs have been successfully used for learning generalized MCDA models from decision examples in a preference disaggregation setting. Wang and Malakooti (1992), and Malakooti and Zhou (1994) used feedforward ANN models to learn an arbitrary value function for ranking a set of alternatives, as well as to learn a relational multicriteria model based on pairwise comparisons (binary relations) among the alternatives. Generalized network decision models have a function free form, which is less restricted by the assumptions imposed in MAVT (Keeney and Raiffa 1993). Experimental simulation results showed that ANN models performed very well in representing various forms of decision models, outperforming other popular model development techniques based on linear programming formulations. Wang *et al*. (1994) applied a similar ANN model to a job shop production system problem.

In a different framework compared with the aforementioned studies, Stam *et al*. (1996) used ANNs within the context of the analytic hierarchy process (AHP; Saaty 2006). AHP is based on a hierarchical structuring of the decision problem, with the overall goal on the top of the hierarchy and the alternatives at the bottom. With this hierarchical structure, the DM is asked to perform pairwise comparisons of the elements at each level of the hierarchy with respect to the elements of the preceding (higher) level. Stam *et al*. investigated two different ANN structures for accurately approximating the preferences ratings of the alternatives, within the context of imprecise preference judgments by the DM. They showed that a modified Hopfield network has very close connections to the mechanics of the AHP, but found that this network formulation cannot provide good results in estimating the mapping from a positive reciprocal pairwise comparison matrix to its preference rating vector. On the other hand, a feed-forward ANN model was found to provide very good approximations of the preference ratings in the presence of impreciseness. This ANN model was actually superior to the standard principal eigenvector method.

Similar ANN-based methodologies have also be used to address dynamic MCDA problems (where the DM's preferences change over time; Malakooti and Zhou 1991), to learn fuzzy preferences (Wang 1994a,b; Wang and Archer 1994) and outranking relations

(Hu 2009), to provide support in group decision-making problems (Wang and Archer 1994), as well as in multicriteria clustering (Malakooti and Raman 2000).

ANNs have also been employed for preference representation and learning in multiobjective optimization. Within this context, Sun *et al*. (1996) proposed a feed-forward ANN model, which is trained to represent the DM's preference structure. The trained ANN model serves as a value function, which is maximized in order to identify the efficient solution that best fits the DM's preferences. Sun *et al*. (2000) used a similar feed-forward ANN approach to facilitate the interactive solution process in multiobjective optimization problems. Other ANN architectures have also been used as multiobjective optimizers (Gholamian *et al*. 2006; McMullen 2001) and hybrid evaluation systems (Raju *et al*. 2006; Sheu 2008).

A comprehensive overview of the contributions of ANNs in MCDA in provided by Hanne in Chapter 5.

1.3.1.2 Rule-based models

Rule-based and decision tree models are very popular within the machine learning research community. The symbolic nature of such models makes them easy to understand, which is important in the context of decision aiding. During the last decade significant research has been devoted to the use of such approaches as preference modeling tools in MCDA.

In particular, a significant part of the research related to the use of rule-based models in MCDA has focused on rough set theory (Pawlak 1982; Pawlak and Słowiński 1994), which provides a complete and well-axiomatized methodology for constructing decision rule preference models from decision examples. Rough sets have been initially introduced as a methodology to describe dependencies between attributes, to evaluate the significance of attributes and to deal with inconsistent data in the context of machine learning. However, significant research has been conducted on the use of the rough set approach as a methodology for preference modeling in multicriteria decision problems (Greco *et al*. 1999, 2001). The decision rule models developed through the rough set approach for MCDA problems are built on the basis of the dominance relation. Each 'if... then...' decision rule is composed of a condition part specifying a partial profile on a subset of criteria to which an alternative is compared using the dominance relation, and a conclusion part suggesting a decision recommendation.

Decision rule preference models have been initially developed in the context of multicriteria classification problems. In this case the recommendations in the conclusion part of each rule involve the assignment of the alternatives either in a specific class or a set of classes. Extensions to ranking and choice decision problems have been developed by Greco *et al*. (2001) and Fortemps *et al*. (2008), whereas Greco *et al*. (2008a) presented a dominance-based rough set approach for multiobjective optimization.

The decision rule preference model has also been considered in terms of conjoint measurement (Greco *et al*. 2004) and Bayesian decision theory (Greco *et al*. 2007). Greco *et al*. (2004) showed that there is an equivalence of simple cancellation property, a general discriminant function and a specific outranking relation, on the one hand, and the decision rule model on the other hand. They also showed that the decision rule model resulting from the dominance-based rough set approach has an advantage over the usual functional and relational models because it permits the handling of inconsistent

decision instances. Inconsistency decision instances often appear due to the instability of preferences, the incomplete determination of criteria, and the hesitation of the DM.

In Chapter 6, Szeląg *et al*. provide a comprehensive presentation of rule-based decision models, focusing on MCDA ranking problems.

1.3.1.3 Kernel methods

Kernel methods are widely used for pattern classification, regression analysis, and density estimation. Kernel methods map the problem data to a high dimensional space (feature space), thus enabling the development of complex nonlinear decision and prediction models, using linear estimation methods (Schölkopf and Smola 2002). The data mapping process is implicitly defined through the introduction of (positive definite) kernel functions. Support vector machines (SVMs) are the most widely used class of kernel methods. Recently, they have also been used within the context of preference learning for approximating arbitrary utility/value functions and preference aggregation.

Herbrich *et al*. (2000) illustrated the use of kernel approaches, within the context of SVM formulations, for representing value/ranking functions of the generalized form $V(\mathbf{x}) = \mathbf{w}\phi(\mathbf{x})$, where ϕ is a possibly infinite-dimensional and in general unknown feature mapping. The authors derived bounds on the generalizing performance of the estimated ranking models, based on the margin separating objects in consecutive ranks.

Waegeman *et al*. (2009) extended this approach to relational models. In this case, the preference model of the form $f(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{w}\phi(\mathbf{x}_i, \mathbf{x}_j)$ is developed to represent the preference of alternative i compared with alternative j. This framework is general enough to accommodate special modeling forms. For instance, it includes value models as a special case, and similar techniques can also be used to kernelize Choquet integrals. As an example, Waegeman *et al*. illustrated the potential of this framework in the case of valued concordance relations, which are used in the ELECTRE methods.

Except for the development of generalized decision models, kernel methods have also been employed for robust model inference purposes. For instance, Evgeniou *et al*. (2005) showed how the regularization principle (which is at the core of the theory of kernel methods) is related to the robust fitting of linear and polynomial value function models in ordinal regression problems. Doumpos and Zopounidis (2007) employed the same regularization principle for developing new improved linear programming formulations for fitting additive value functions in ranking and classification problems. The development of additive value function was also addressed by Dembczynski *et al*. (2006) who presented a methodology integrating the dominance-based rough set approach and SVMs.

SVMs have also be used in the context of multiobjective optimization (Aytug and Sayin 2009; Yun *et al*. 2009) in order to approximate the set of Pareto optimal solutions in complex nonlinear problems. Multiobjective and goal programming formulations has also been used for training SVM models (Nakayama and Yun 2006; Nakayama *et al*. 2005). Finally, hybrid systems based on SVMs have been proposed. For instance, Jiao *et al*. (2009) combined SVMs with the UTADIS disaggregation method (Doumpos and Zopounidis 2002a) for the development of accurate multi-group classification models.

1.3.2 Fuzzy modeling

Decision making is often based on fuzzy, ambiguous, and vague judgments. Verbal expressions such as ‘almost,’ ‘usually,’ ‘often,’ etc., are simple yet typical examples

of the ambiguity and vagueness often encountered in the decision-making process. The fuzzy set theory first introduced by Zadeh (1965), provides the necessary modeling tools for such situations. The concept of a fuzzy set is at the center of this approach. In the traditional set theory, a set is considered as a collection of well defined and distinct objects, which implies that sets have clearly defined (crisp) boundaries. Therefore, a statement of the form 'object x belongs to set A' is either true or false. On other hand, a fuzzy set has no crisp boundaries, and every object is associated with a degree of membership with respect to a fuzzy set.

Since its introduction, fuzzy set theory has been an extremely active research field with numerous practical applications in engineering and management. Its uses in the context of decision aiding have also attracted much interest.

1.3.2.1 Fuzzy multiobjective optimization

The traditional multiobjective programming framework assumes that all the parameters of the problem are well-defined. However, imprecision, vagueness, and uncertainty can make the specification of goals, targets, objectives, and constraints troublesome and unclear. Bellman and Zadeh (1970) were the first to explore optimization models in the context of fuzzy set theory. Zimmermann (1976, 1978) further investigated this idea both in the case of single-objective problems as well as in the context of multiobjective optimization.

Fuzzy multiobjective programming formulations have a similar form to conventional multiobjective programming problems (i.e., the optimization of several objective functions over some constraints). The major distinction between these two approaches is that while in deterministic multiobjective programming all objective functions and constraints are specified in a crisp way, in fuzzy multiobjective programming they are specified using the fuzzy set theory through the introduction of membership functions. Fuzzy coefficients for the decision variables in the objective function and the constraints can also be introduced.

A major advantage of fuzzy multiobjective programming techniques over conventional mathematical programming with multiple objectives, is that it provides a framework to address optimization problems within a less strict context regarding the sense of the imposed constraints, as well as the degree of satisfaction of the DM from the compromise solutions that are obtained (i.e., introduction of fuzzy objectives).

The FLIP method (Słowiński 1990) for multiobjective linear programming problems is a typical example of the integration of the fuzzy set theory with multiobjective optimization techniques. FLIP considers uncertainty through the definition of all problem parameters (objective function coefficients, variables' coefficients in the constraints, right-hand side coefficients) as fuzzy numbers, each one associated with a possibility distribution. The recent book by Sakawa *et al*. (2011) presents a comprehensive overview of the theory of fuzzy multiobjective programming including stochastic problems, whereas Roubens and Teghem (1991) present a survey of fuzzy multiobjective programming and stochastic multiobjective optimization and perform a comparative investigation of the two areas.

A detailed presentation of the principles and techniques for fuzzy multiobjective optimization is presented by Sakawa in Chapter 10.

1.3.2.2 Fuzzy preference modeling

Preference modeling is a major research topic in MCDA. The modeling of a DM's preferences can be viewed within the context of MAVT models as well as in the context of

outranking relations (Fodor and Roubens 1994; Roubens 1997). The concept of outranking relation is closely connected with the philosophy of fuzzy sets. For instance, in ELECTRE methods the outranking relation $\mathbf{x}_i \; S \, \mathbf{x}_j$ is constructed to evaluate whether alternative i is at least as good as alternative j. Similarly, in the PROMETHEE methods a preference relation is constructed to measure the preference for alternative i over alternative j. In both sets of methods the outranking/preference relations are not treated in a crisp setting. Instead, the relations are quantified by proper measures (e.g., credibility index in ELECTRE and preference index in PROMETHEE) representing the strength of the outranking/preference of one alternative over another. For instance, the credibility index $\sigma(\mathbf{x}_i, \mathbf{x}_j)$ used in ELECTRE methods represents the validity of the affirmation 'alternative i outranks alternative j.' Thus, it is a form of membership function. Perny and Roy (1992) provided a comprehensive discussion on the use of fuzzy outranking relations in preference modeling together with an analysis of the characteristics and properties of such relations.

Despite the above fundamental connection between commonly used MCDA outranking techniques and fuzzy theory, it should be noted that traditional outranking methods consider crisp data. However, many extensions for handling fuzzy data in outranking methods have been proposed. For instance, Czyzak and Słowiński (1996) considered the evaluations of the alternatives on the criteria as fuzzy numbers in order to construct an outranking relation. Common aggregation operators (e.g., maximum and minimum) are employed to aggregate these fuzzy numbers in order to perform the necessary concordance and discordance tests similarly to the traditional outranking relations approach. Roubens (1996) presented several procedures for aggregating fuzzy criteria in an outranking context for choice and ranking problem, whereas a more recent overview of this research direction is given by Bufardi *et al*. (2008). Fuzzy relations can also be used to handle the fuzziness that characterize the DM's preferences. For instance, Siskos (1982) proposed a methodology using disaggregation techniques to build a fuzzy outranking relation on the basis of the information represented in multiple additive value functions which are compatible with the DM's preferences, thus modeling the DM's fuzzy preferential system.

Fuzzy preference modeling approaches have also been developed in the context of MAVT. Grabisch (1995; 1996) introduced an approach to manage uncertainty in the MAVT framework through the consideration of the concept of fuzzy integrals initially introduced by Sugeno (1974). In the proposed approach fuzzy integrals are used instead of the additive and multiplicative aggregation operators that are commonly used in MAVT in order to aggregate all attributes into a single evaluation index (value function). The major advantageous feature of employing fuzzy integrals within the MAVT context is their ability to consider the interactions among the evaluation criteria including redundancy and synergy. On the other hand, the major drawback of such an approach that is a consequence of its increased complexity over simple aggregation procedures (e.g., weighted average), involves the increased number of parameters that should be defined, either directly by the DM, or employing heuristics and optimization techniques. The use of the Choquet integral as an aggregation function has also attracted much interest among MCDA researchers. Marichal and Roubens (2000) first introduced a methodology implementing this approach in a preference disaggregation context. Some work on this topic can be found in the papers of Angilella *et al*. (2004; 2010) and Kojadinovic (2004; 2007), while a review of this

area is given in the paper by Grabisch *et al*. (2008). Other applications of the fuzzy set theory to MAVT are discussed in the book by Lootsma (1997).

A final class of decision models developed within the context of fuzzy set theory that attracted much interest in the context of MCDA is based on the ordered weighted averaging (OWA) approach first introduced by Yager (1988). An OWA aggregation model, is a particular case of the Choquet integral, which is similar to a simple weighted average model. However, instead of weighting the criteria, an OWA model assigns weights to the relative position of one criterion value with respect to the other values (Torra 2010). In this way, OWA models allow for different compensation levels to be modeled. For instance, assigning high weights to low performances lead to a noncompensatory mode, whereas compensation can be allowed if higher weight is given to good performance levels. In the context of decision making under uncertainty, the OWA aggregation scheme is a generalization of the Hurwicz rule. Yager (1993) and Xu and Da (2003) provide overviews of different OWA models, whereas Yager (2004) extends this framework to consider different criteria priorities in an MCDA context.

1.3.3 Metaheuristics

Metaheuristics have been one of the most active and rapidly evolving fields in computational intelligence and operations research. Their success and development is due to the highly complex nature of many decision problems. As a consequence the corresponding mathematical models are nonlinear, nonconvex, and/or combinatorial in nature, thus making it very difficult to solve them through traditional optimization algorithms. Metaheuristics and evolutionary techniques have been very successful in dealing with computationally intensive optimization problems, as they make few or no assumptions about the problem and can search very large solution spaces very efficiently. In the context of MCDA, such methods have been primarily used for multiobjective optimization. Their use for fitting complex decision models in a preference disaggregation setting has also attracted some interest.

1.3.3.1 Evolutionary methods and metaheuristics in multiobjective optimization

Traditional multiobjective optimization techniques seek to find an efficient solution that best fits the preferences of a DM. The solution process is performed iteratively so that the DM's preferences are progressively specified and refined until the most satisfactory solution is identified. During this process a series of optimization problems needs to be solved, which may not be easy in the case of combinatorial or highly complex nonlinear and nonconvex problems. Furthermore, in such procedures the DM is often not provided with a full idea of the whole set of Pareto optimal solutions. Metaheuristics are well-suited in this context as they are applicable in all types of computationally intensive multiobjective optimization problems and enable the approximation of complex Pareto sets in a single run of an algorithmic procedure.

Different classes of algorithms can be identified in this research direction. Approaches based on genetic algorithms (GAs) are probably the most popular. GAs are computational procedures that mimic the process of natural evolution for solving complex optimization problems. They implement stochastic search schemes to evolve an initial population (set) of solutions through selection, mutation, and crossover operators until a good solution is reached. The first GA-based approach for multiobjective optimization problem was

proposed by Schaffer (1985). During the 1990s and the 2000s many other algorithms implementing a similar GA approach have been proposed. A comprehensive presentation of this approach can be found in the book by Deb (2001), whereas Konak *et al*. (2006) presented a tutorial and review of the field.

The differential evolution (DE) algorithm has also attracted much interest for multiobjective optimization. DE has been introduced by Storn and Price (1997) as a powerful alternative to GAs, which is well-suited to continuous optimization problems. Similarly to a GA, DE also employs evolution operators to evolve a generation of solutions, but it is based on greedy search strategies, which ensure that solutions are strictly improved in every iteration of the algorithm. Abbass and Sarker (2002) presented one of the first implementations of the DE scheme in multiobjective optimization. Some recent extensions have been presented by Gong *et al*. (2009), Krink and Paterlini (2011), and Wang and Cai (2012), whereas Mezura-Montes *et al*. (2008) present a review of DE-based multiobjective optimization algorithms.

A third class of computational intelligence techniques for solving multiobjective optimization problems involves other metaheuristic algorithms, such as simulated annealing, tabu search, ant colony optimization, and particle swarm optimization, which have been proved very successful in solving complex optimization problems of a combinatorial nature. The use of such algorithms in multiobjective optimization can be found in Landa Silva *et al*. (2004), Molina *et al*. (2007), Bandyopadhyay *et al*. (2008), Doerner *et al*. (2008), and Elhossini *et al*. (2010). Jones *et al*. (2002) present an overview of the field, whereas Ehrgott and Gandibleux (2008) focus on recent approaches, where metaheuristics are combined with exact methods.

In Chapter 8, Jaimes and Coello Coello present in detail different interactive methods for multiobjective optimization.

1.3.3.2 Preference disaggregation with evolutionary techniques

Inferring simple decision-making models (e.g., additive or linear value functions) from decision examples poses little computational problems. Most existing preference disaggregation techniques use linear programming for this purpose (Jacquet-Lagrèze and Siskos 2001; Zopounidis and Doumpos 2002). However, more complex models cannot be constructed with exact methods. Metaheuristics are well-suited in this context and have attracted the interest of MCDA researchers over the past few years.

Most of the research on this area has focused on outranking models. Goletsis *et al*. (2004) used a GA for the development of an outranking model based on the philosophy of the ELECTRE methods in a medical classification problem. Belacel *et al*. (2007) used the reduced variable neighborhood search metaheuristic to infer the parameters of the PROAFTN outranking method from a set of reference examples. Focusing on the same outranking method Al-Obeidat *et al*. (2011) used a particle swarm optimization algorithm. Fernandez *et al*. (2009) developed a model based on a fuzzy indifference relation for classification purposes. In order to infer the parameters of the model from a set of reference examples they used the NSGA-II multiobjective evolutionary algorithm (Deb *et al*. 2002) considering four measures related to the inconsistencies and the correct recommendations of the decision model. A similar approach was also presented by Fernandez and Navarro (2011). Doumpos *et al*. (2009) presented a methodology based on the differential evolution algorithm for estimating all the parameters of an ELECTRE TRI model from assignment examples in classification problems under both the optimistic and

the pessimistic assignment rules (Roy and Bouyssou 1993). Doumpos and Zopounidis (2011a) applied this methodology to a large data set for the development of credit rating models and demonstrated how the special features of ELECTRE TRI can provide useful insights into the relative importance of the credit rating criteria and the characteristics of the alternatives. Eppe *et al*. (2011) employed the NSGA-II algorithm for inferring the parameters of PROMETHEE II models from decision instances. The authors suggested a bi-objective approach according to which the model is developed so that the number of inconsistencies compared with the DM's evaluation of the reference alternatives is minimized and the robustness of the model's parameters estimates is maximized. In contrast to all the aforementioned studies, which focused on outranking models, Doumpos (2012) considered the construction of a nonmonotone additive value function, assuming that the marginal value functions are quasi-convex. The differential evolution algorithm was used to infer the additive function from reference examples in a classification setting.

1.4 Conclusions

In a dynamic environment characterized by increasing complexity and considerable uncertainties, the interdisciplinary character of decision analysis and decision aiding is strengthened. Complex and ill-structured decision problems in engineering and management cannot be handled in a strictly defined methodological context. Instead, integrated approaches often need to be implemented, combining concepts and techniques from various research fields. In this context, the relationship between artificial intelligence and MCDA has attracted much interest among decision scientists.

This chapter presented an overview of this area, focusing on the computational intelligence paradigm. Computational intelligence has been one of the most active areas in artificial intelligence research, with numerous applications engineering and management systems. The overview focused on the contributions of computational intelligence methodologies in decision support, covering important issues such as the introduction of new preference modeling techniques, advanced algorithmic solution procedures for complex problems, as well as new techniques for constructing decision models. The advances in each of these areas provides new capabilities for extending the research and practice of the MCDA paradigm, thus enabling its use in new ill-structured decision domains, characterized by uncertainty, vagueness, and imprecision, complex preference and data structures, and high data dimensionality.

The active research conducted in this area is expected to continue to grow at a rapid pace. Future research could cover various issues. For instance, the integration with other artificial intelligence paradigms, including knowledge management, representation, and engineering, natural language processing, intelligent agents (see Chapter 11), evidential reasoning (see Chapter 7) and Bayesian inference, is an interesting research area providing a wide range of new potentials for enhancing decision aiding. The implementation of the state-of-the-art research results into intelligent decision support systems taking advantage of the rapid advances in data management and web-based technologies is also important for disseminating new results among researchers and improving their applicability in practice (see Chapter 2). Finally, given the wide arsenal of existing approaches, their comprehensive comparative empirical evaluation is necessary in order to identify their strengths, weaknesses, and limitations, under different decision aiding settings.

References

Abbass HA and Sarker RA (2002) The Pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools* **11**(4), 531–552.

Al-Obeidat F, Belacel N, Carretero JA and Mahanti P (2011) An evolutionary framework using particle swarm optimization for classification method PROAFTN. *Applied Soft Computing* **11**(8), 4971–4980.

Angilella S, Greco S, Lamantia F and Matarazzo B (2004) Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research* **158**(3), 734–744.

Angilella S, Greco S and Matarazzo B (2010) Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research* **201**(1), 277–288.

Aouni B and Kettani O (2001) Goal programming model: A glorious history and a promising future. *European Journal of Operational Research* **133**, 225–231.

Aytug H and Sayin S (2009) Using support vector machines to learn the efficient set in multiple objective discrete optimization. *European Journal of Operational Research* **193**(2), 510–519.

Bandyopadhyay S, Saha S, Maulik U and Deb K (2008) A simulated annealing based multi-objective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation* **12**(3), 269–283.

Belacel N, Bhasker Raval H and Punnenc AP (2007) Learning multicriteria fuzzy classification method PROAFTN from data. *Computers and Operations Research* **34**, 1885–1898.

Bellman R and Zadeh LA (1970) Decision making in a fuzzy environment. *Management Science* **17**, 141–164.

Bouyssou D (1990) Building criteria: A prerequisite for MCDA. In *Readings in Multiple Criteria Decision Aid* (ed. Bana e Costa CA). Springer, Berlin, pp. 27–71.

Brans JP and Mareschal B (2005) PROMETHEE methods. In *Multiple Criteria Decision Analysis-State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, Boston, pp. 163–195.

Breiman L (2001) Statistical modeling: The two cultures. *Statistical Science* **16**(3), 199–231.

Bufardi A, Gheorghe R and Xirouchakis P (2008) Fuzzy outranking methods: Recent developments. In *Fuzzy Multi-Criteria Decision Making* (ed. Kahraman C). Springer, New York, pp. 119–157.

Craenen BGW and Eiben AE (2003) Computational intelligence. In *Encyclopedia of Life Support Sciences*. EOLSS Publishers. Available at http://bcraenen.home.xs4all.nl/publications/eolss.pdf.

Czyzak P and Słowiński R (1996) Possibilistic construction of fuzzy outranking relation for multiple-criteria ranking. *Fuzzy Sets and Systems* **81**(1), 123–131.

Deb K (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, New York.

Deb K, Agrawal S, Pratap A and Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Dembczynski K, Kotlowski W and Słowiński R (2006) Additive preference model with piecewise linear components resulting from dominance-based rough set approximations. In *Artificial Intelligence and Soft Computing–ICAISC 2006* (eds Rutkowski L, Tadeusiewicz R, Zadeh LA and Zurada J), vol. 4029 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, pp. 499–508.

Dias L, Mousseau V, Figueira J and Clímaco J (2002) An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI. *European Journal of Operational Research* **138**(2), 332–348.

Doerner KF, Gutjahr WJ, Hartl RF, Strauss C and Stummer C (2008) Nature-inspired metaheuristics for multiobjective activity crashing. *Omega* **36**(6), 1019–1037.

Doumpos M (2012) Learning non-monotonic additive value functions for multicriteria decision making. *OR Spectrum* **34**(1), 89–106.

Doumpos M, Marinakis Y, Marinaki M and Zopounidis C (2009) An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method. *European Journal of Operational Research* **199**(2), 496–505.

Doumpos M and Zopounidis C (2002a) *Multicriteria Decision Aid Classification Methods*. Springer, New York.

Doumpos M and Zopounidis C (2002b) On the development of an outranking relation for ordinal classification problems: An experimental investigation of a new methodology. *Optimization Methods and Software* **17**(2), 293–317.

Doumpos M and Zopounidis C (2004) A multicriteria classification approach based on pairwise comparisons. *European Journal of Operational Research* **158**, 378–389.

Doumpos M and Zopounidis C (2007) Regularized estimation for preference disaggregation in multiple criteria decision making. *Compututational Optimization and Applications* **38**, 61–80.

Doumpos M and Zopounidis C (2011a) A multicriteria outranking modeling approach for credit rating. *Decision Sciences* **42**(3), 721–742.

Doumpos M and Zopounidis C (2011b) Preference disaggregation and statistical learning for multicriteria decision support: A review. *European Journal of Operational Research* **209**(3), 203–214.

Duch W (2007) What is computational intelligence and where is it going? In *Challenges for Computational Intelligence* (eds Duch W and Mandziuk K), vol. 63 of Studies in Computational Intelligence. Springer, Berlin, pp. 1–13.

Ehrgott M (2005) *Multicriteria Optimization*, 2nd edn. Springer, Berlin.

Ehrgott M and Gandibleux X (2002) *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*. Springer, Berlin.

Ehrgott M and Gandibleux X (2008) Hybrid metaheuristics for multi-objective combinatorial optimization. In *Hybrid Metaheuristics - An Emerging Approach to Optimization* (eds Blum C, Aguilera MJ, Roli A and Sampels M). Springer-Verlag, Berlin, pp. 221–259.

Elhossini A, Areibi S and Dony R (2010) Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evolutionary Computation* **18**(1), 127–156.

Engelbrecht AP (2002) *Computational Intelligence: An Introduction*. John Wiley & Sons, Ltd, Chichester.

Eppe S, De Smet Y and Stützle T (2011) A bi-objective optimization model to eliciting decision maker's preferences for the PROMETHEE II method. In *Algorithmic Decision Theory* (eds Brafman RI, Roberts FS and Tsoukias A), vol. 6992 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, pp. 56–66.

Evgeniou T, Boussios C and Zacharia G (2005) Generalized robust conjoint estimation. *Marketing Science* **24**(3), 415–429.

Fernandez E and Navarro J (2011) A new approach to multi-criteria sorting based on fuzzy outranking relations: The THESEUS method. *European Journal of Operational Research* **213**(2), 405–413.

Fernandez E, Navarro J and Bernal S (2009) Multicriteria sorting using a valued indifference relation under a preference disaggregation paradigm. *European Journal of Operational Research* **198**(2), 602–609.

Fodor J and Roubens M (1994) *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer Academic Publishers, Dordrecht.

Fortemps P, Greco S and Słowiński R (2008) Multicriteria decision support using rules that represent rough-graded preference relations. *European Journal of Operational Research* **188**, 206–223.

Gholamian MR, Fatemi Ghomi SMT and Ghazanfari M (2006) A hybrid intelligent system for multiobjective decision making problems. *Computers and Industrial Engineering* **51**, 26–43.

Goletsis Y, Papaloukas C, Fotiadis D, Likas A and Michalis L (2004) Automated ischemic beat classification using genetic algorithms and multicriteria decision analysis. *IEEE Transactions on Biomedical Engineering* **51**(10), 1717–1725.

Gong W, Cai Z and Zhu L (2009) An efficient multiobjective differential evolution algorithm for engineering design. *Structural and Multidisciplinary Optimization* **38**(2), 137–157.

Grabisch M (1995) Fuzzy integrals in multicriteria decision making. *Fuzzy Sets and Systems* **69**, 279–298.

Grabisch M (1996) The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research* **89**, 445–456.

Grabisch M, Kojadinovic I and Meyer P (2008) A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package. *European Journal of Operational Research* **186**(2), 766–785.

Greco S, Matarazzo B and Słowiński R (1999) Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research* **117**, 63–83.

Greco S, Matarazzo B and Słowiński R (2001) Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research* **129**, 1–47.

Greco S, Matarazzo B and Słowiński R (2004) Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research* **158**(2), 271–292.

Greco S, Matarazzo B and Słowiński R (2008a) Dominance-based rough set approach to interactive multiobjective optimization. In *Multiobjective Optimization* (eds Branke J, Deb K, Miettinen K and Słowiński R). Springer, Berlin, pp. 121–155.

Greco S, Mousseau V and Słowiński R (2008b) Ordinal regression revisited: Multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research* **191**(2), 415–435.

Greco S, Słowiński R and Yao Y (2007) Bayesian decision theory for dominance-based rough set approach. In *Rough Sets and Knowledge Technology* (eds Yao J, Lingras P, Wu WZ, Szczuka M, Cercone N and Ślęzak S). Springer, Berlin, pp. 134–141.

Hand D, Mannila H and Smyth P (2001) *Principles of Data Mining*. MIT Press, Cambridge.

Herbrich R, Graepel T and Obermayer K (2000) Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers* (eds Smola AJ, Bartlett PL, Schölkopf B and Schuurmans D). MIT Press, Cambridge, MA, pp. 115–132.

Hu YC (2009) Bankruptcy prediction using ELECTRE-based single-layer perceptron. *Neurocomputing* **72**, 3150–3157.

Jacquet-Lagrèze E and Siskos Y (1982) Assessing a set of additive utility functions for multicriteria decision making: The UTA method. *European Journal of Operational Research* **10**, 151–164.

Jacquet-Lagrèze E and Siskos Y (2001) Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research* **130**, 233–245.

Jiao T, Peng J and Terlaky T (2009) A confidence voting process for ranking problems based on support vector machines. *Annals of Operations Research* **166**(1), 23–38.

Jones DF, Mirrazavi SK and Tamiz M (2002) Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research* **137**(1), 1–9.

Jones DF and Tamiz M (2002) Goal programming in the period 1990–2000. In *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys* (eds Ehrgott M and Gandibleux X). Springer, Berlin, pp. 129–170.

Jones DF and Tamiz M (2010) *Practical Goal Programming*. Springer, New York.

Keeney RL and Raiffa H (1993) *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Cambridge University Press, Cambridge.

Kojadinovic I (2004) Estimation of the weights of interacting criteria from the set of profiles by means of information-theoretic functionals. *European Journal of Operational Research* **155**(3), 741–751.

Kojadinovic I (2007) Minimum variance capacity identification. *European Journal of Operational Research* **177**(1), 498–514.

Konak A, Coit DW and Smith AE (2006) Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety* **91**(9), 992–1007.

Krink T and Paterlini S (2011) Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science* **8**(1–2), 157–179.

Landa Silva JD, Burke EK and Petrovic S (2004) An introduction to multiobjective metaheuristics for scheduling and timetabling. In *Metaheuristics for Multiobjective Optimisation* (eds Gandibleux X, Sevaux M, Sörensen K and T'Kindt V), vol. 535 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, pp. 91–129.

Lootsma FA (1997) *Fuzzy Logic for Planning and Decision Making*. Kluwer Academic Publishers, Dordrecht.

Malakooti B and Raman V (2000) Clustering and selection of multiple criteria alternatives using unsupervised and supervised neural networks. *Journal of Intelligent Manufacturing* **11**, 435–451.

Malakooti B and Zhou Y (1991) A recursive ann for solving adaptive multiple criteria problems. *Pure Mathematics and Applications Series C* **2**(2–4), 165–176.

Malakooti B and Zhou YQ (1994) Feedforward artificial neural networks for solving discrete multiple criteria decision making problems. *Management Science* **40**(11), 1542–1561.

Marichal JL and Roubens M (2000) Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research* **124**(3), 641–650.

McMullen PR (2001) A kohonen self-organizing map approach to addressing a multiobjective mixed-model JIT sequencing problem. *International Journal of Production Economics* **72**, 59–71.

Mezura-Montes E, Reyes-Sierra M and Coello Coello CA (2008) Multi-objective optimization using differential evolution: A survey of the state-of-the-art. In *Advances in Differential Evolution* (ed. Chakraborty UK). Springer, Berlin, pp. 173–196.

Miettinen K (1998) *Nonlinear Multiobjective Optimization*. Kluwer. Academic Publishers, Dordrecht.

Molina J, Laguna M, Martí R and Caballero R (2007) SSPMO: A scatter tabu search procedure for non-linear multiobjective optimization. *INFORMS Journal on Computing* **19**(1), 91–100.

Mousseau V, Figueira J and Naux JP (2001) Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. *European Journal of Operational Research* **130**, 263–275.

Nakayama H and Yun Y (2006) Support vector regression based on goal programming and multi-objective programming. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006*. IEEE Press, pp. 1156–1161.

Nakayama H, Yun Y, Asada Y and Yoon M (2005) MOP/GP models for machine learning. *European Journal of Operational Research* **166**(3), 756–768.

Pardalos PM, Siskos Y and Zopounidis C (1995) *Advances in Multicriteria Analysis*. Kluwer Academic Publishers, Dordrecht.

Pawlak Z (1982) Rough sets. *International Journal of Information and Computer Sciences* **11**, 341–356.

Pawlak Z and Słowiński R (1994) Rough set approach to multi-attribute decision analysis. *European Journal of Operational Research* **72**, 443–459.

Perny P and Roy B (1992) The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems* **49**, 33–53.

Raju KS, Kumar DN and Duckstein L (2006) Artificial neural networks and multicriterion analysis for sustainable irrigation planning. *Computers and Operations Research* **33**, 1138–1153.

Roubens M (1996) Choice procedures in fuzzy multicriteria decision analysis based on pairwise comparisons. *Fuzzy Sets and Systems* **84**(2), 135–142.

Roubens M (1997) Fuzzy sets and decision analysis. *Fuzzy Sets and Systems* **90**(2), 199–206.

Roubens M and Teghem J (1991) Comparison of methodologies for fuzzy and stochastic multiobjective programming. *Fuzzy Sets and Systems* **42**(1), 119–132.

Roy B (1968) Classement et choix en présence de points de vue multiples: La méthode ELECTRE. *La Revue d'Informatique et de Recherche Opérationelle* **8**, 57–75.

Roy B (1985) *Méthodologie Multicritère d'Aide à la Décision*. Economica, Paris.

Roy B (1991) The outranking approach and the foundations of ELECTRE methods. *Theory and Decision* **31**, 49–73.

Roy B and Bouyssou D (1993) *Aide MulticritÃÍre à la Décision: Méthodes et Cas*. Economica, Paris.

Saaty TL (2006) *Fundamentals of the Analytic Hierarchy Process*. RWS Publications, Pittsburgh, PA.

Sakawa M, Nishizaki I and Katagiri H (2011) *Fuzzy Stochastic Multiobjective Programming*. Springer, New York.

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms* Lawrence Erlbaum Associates Inc., Hillsdale, NJ, pp. 93–100.

Schölkopf B and Smola A (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA.

Sheu JB (2008) A hybrid neuro-fuzzy analytical approach to mode choice of global logistics management. *European Journal of Operational Research* **189**(3), 971–986.

Siskos J (1982) A way to deal with fuzzy preferences in multicriteria decision problems. *European Journal of Operational Research* **10**(3), 314–324.

Siskos J, Grigoroudis E and Matsatsinis N (2005) UTA methods. In *Multiple Criteria Decision Analysis–State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, Boston, pp. 297–343.

Słowiński R (1990) FLIP - an interactive method for multiobjective linear programming with fuzzy coefficients. In *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty* (eds Słowiński R and Teghem J). Kluwer Academic Publishers, Dordrecht, pp. 249–262.

Stam A, Sun M and Haines M (1996) Artificial neural network representations for hierarchical preference structures. *Computers and Operations Research* **23**(12), 1191–1201.

Steuer R (1985) *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, Ltd, New York.

Storn R and Price K (1997) Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341–359.

Sugeno M (1974) *Theory of Fuzzy Integrals and its Applications*. PhD thesis, Tokyo Institute of Technology, Tokyo.

Sun M, Stam A and Steuer RE (1996) Solving multiple objective programming problems using feed-forward artificial neural networks: The interactive FFANN procedure. *Management Science* **42**(6), 835–849.

Sun M, Stam A and Steuer RE (2000) Interactive multiple objective programming using Tchebycheff programs and artificial neural networks. *Computers and Operations Research* **27**(7–8), 601–620.

Torra V (2010) Aggregation operators for evaluating alternatives. In *Fuzzy Optimization* (eds Lodwick W and Kacprzyk J), vol. 254 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin, pp. 65–76.

Waegeman W, De Baets B and Boullart B (2009) Kernel-based learning methods for preference aggregation. *4OR* **7**, 169–189.

Wang J (1994a) A neural network approach to modeling fuzzy preference relations for multiple criteria decision making. *Computers and Operations Research* **21**(9), 991–1000.

Wang J (1994b) A neural network approach to multiple criteria decision making based on fuzzy preference information. *Information Sciences* **78**, 293–302.

Wang J and Malakooti B (1992) A feedforward neural network for multiple criteria decision making. *Computers and Operations Research* **19**(2), 151–167.

Wang J, Yang JQ and Lee H (1994) Multicriteria order acceptance decision support in over-demanded job shops: A neural network approach. *Mathematical and Computer Modelling* **19**(5), 1–19.

Wang S and Archer NP (1994) A neural network technique in modeling multiple criteria multiple person decision making. *Computers and Operations Research* **21**(2), 127–142.

Wang Y and Cai Z (2012) Combining multiobjective optimization with differential evolution to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation* **16**(1), 117–134.

Xu ZS and Da QL (2003) An overview of operators for aggregating information. *International Journal of Intelligent Systems* **18**(9), 953–969.

Yager RR (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Transactions on Systems, Man and Cybernetics* **18**, 183–190.

Yager RR (1993) Families of OWA operators. *Fuzzy Sets and Systems* **59**(2), 125–148.

Yager RR (2004) Modeling prioritized multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* **34**(6), 2396–2404.

Yun Y, Yoon M and Nakayama H (2009) Multi-objective optimization based on meta-modeling by using support vector regression. *Optimization and Engineering* **10**(2), 167–181.

Zadeh LA (1965) Fuzzy sets. *Information and Control* **8**, 338–353.

Zimmermann HJ (1976) Description and optimization of fuzzy systems. *International Journal of General Systems* **2**, 209–215.

Zimmermann HJ (1978) Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems* **1**(1), 45–55.

Zopounidis C and Doumpos M (2002) Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research* **138**(2), 229–246.

2

Intelligent decision support systems

Gloria Phillips-Wren
Sellinger School of Business and Management, Loyola University Maryland, USA

2.1 Introduction

Decision making is a fundamental human activity at the center of our interaction with the world. We know that people make both good decisions and poor decisions, and researchers debate the most effective way to assist (or support) people in arriving at a 'good' decision. One way to characterize decisions, to begin to understand how to assist them, is to categorize decisions as structured, semi-structured or unstructured. Structured decision problems have a known optimal solution and, thus, require limited decision support. For example, a decision that involves the shortest route between two points can be solved analytically with an exact solution. Unstructured decision problems have no agreed-upon criteria or solution and rely on the preferences of the decision maker. For example, deciding one's mate could be considered an unstructured decision. In between these two types of problems, there is a wide range of semi-structured problems that generally have some agreed-upon parameters and yet require human input or preferences for a decision within a specific set of criteria. For example, a business decision on whether to expand the company to global markets could be semi-structured. Semi-structured decision problems are thus amenable to decision support, requiring the combination of interaction with the user and analytical methods to develop alternatives based on criteria and optimal solutions. When artificial intelligence (AI) techniques are utilized in the development of alternatives, the resulting systems are referred to as intelligent decision support systems (IDSS).

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

Researchers remind us that a comprehensive understanding of decision making is needed for effective use of, and benefit from, artificial intelligence (Pomerol 1997; Pomerol and Adam 2008). AI attempts to mimic human decision making in some capacity, and advances in AI have shown significant promise in assisting and improving human decision making, particularly in real-time and complex environments. This chapter will review research in decision making and the decision support systems (DSS) that are based on that understanding, along with the concomitant application of AI techniques to create more powerful IDSS.

2.2 Fundamentals of human decision making

An excellent summary of research on human decision making is provided by Pomerol and Adam (2008). They posit that reasoning and recognition are key poles in decision making that are closely linked. 'Good' decisions are generally characterized by reasoning, a distinctly human characteristic of weighing alternatives and selecting a decision based on criteria. Many types of reasoning can be represented by analytical techniques and, as such, can be embedded into IDSS. Not all decisions are analytical, of course. At the other pole, recognition of stimuli can result in an action or decision as a learned response without an identifiable reasoning basis. However, some progress has been made in understanding responses or decisions in some environments where there is not time for analytical reasoning. Interesting research on 'recognition-primed decisions' can be found in Klein (1993) who studied decision making by firefighters and other emergency response personnel. Klein's work emphasized the use of pattern-matching in situations where human experience or immediate response is required for a successful outcome. In such cases, decision support, if it is used, should supply relevant information and rely more on human processing.

Physiologically, the capacity to make decisions is known to reside in the prefrontal lobe of the brain where reasoning takes place. Damage to this area results in irrational decisions and improper assessment of risk (Adolphs *et al*. 1996; Damasio 1994). Decision making is also known to be affected by emotion in the neural system, both consciously and unconsciously. Recent research in IDSS has demonstrated the ability to model affective characteristics such as emotion within decision making (Santos *et al*. 2011), although the effective inclusion of emotion in machine reasoning is an area of future research. Pomerol and Adam (2008) also point out that working memory is a cognitive function necessary to decision making. Since intelligent information processing systems such as IDSS are built on computer technologies of memory, symbolic reasoning, and the ability to capture and interpret stimuli, IDSS have the necessary capabilities for emulating human decision making.

Models of human decision making underpin systems that have been proposed for DSS and IDSS. An early model of decision making was offered by Savage (1954) and is sometimes called the Expected Utility Model or Subjective Expected Utility Model. Savage assumes that if a decision maker starts with a set of states, a set of outcomes, and a preference order of functions from states to outcomes satisfying certain postulates such as transitivity, then the decision maker has a probability on states and a utility function on outcomes such that he/she is maximizing the expected utility for the said probabilities (Halpern *et al*. 2012). Savage's theorem is often used to suggest the action (or decision) which will maximize the expected utility of the decision given a set of known

probabilities for preceding events. In essence, then, Savage advanced a model of decision making under uncertainty (Karni 2005). There are many criticisms of Savage's theorem, particularly the assumption that the decision maker can evaluate all the consequences of an action, requiring knowledge of all future events with their associated probabilities. However, one of Savage's main contributions was to separate events, outcomes, and actions (Pomerol and Adam 2008).

Simon (1955, 1977, 1997) recognized limitations of Savage's theory and advanced a Theory of Bounded Rationality in which decision makers are limited by the information they possess, time to make the decision, and their own cognitive resources. Simon's seminal work (1955) introduced a behavorial view of decision making as a normative process model consisting of three and then later four phases (Simon 1977, 1997): (1) Intelligence, (2) Design, (3) Choice, and (4) Implementation.

A diagram of the four phases in Simon's process of decision making is shown in Figure 2.1. During the Intelligence phase the decision maker develops an understanding of the problem, articulates the decision to be made, and gathers information relevant

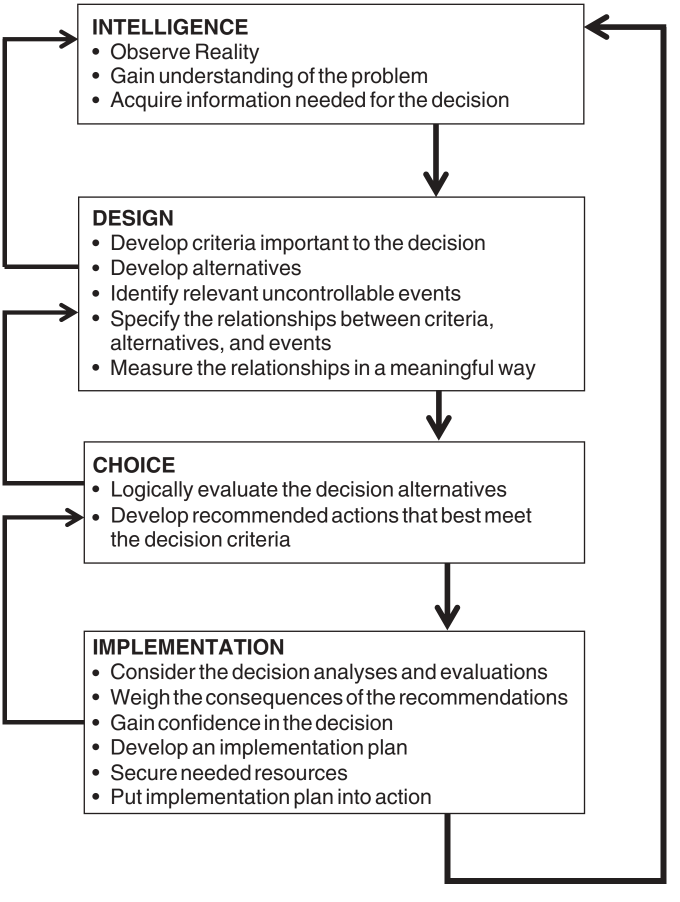

Figure 2.1 Description of Simon's (1977, 1997) model of the process of decision making.

to the decision problem. The Design phase is characterized by identifying variables important to the decision problem, defining criteria for the decision, specifying relationships between variables, developing a decision model that can be used to evaluate alternatives, and exploring alternative decisions. During the Choice phase the decision maker evaluates alternatives and selects a decision that best meets the decision criteria. The final phase of Implementation (sometimes called Review) is the phase in which the decision maker considers the consequences of the decision, develops an implementation plan, secures resources, and acts on the decision. The phases proceed generally sequentially, although there are feedback loops between phases so that the decision maker may return to an earlier phase or begin another decision process based on results from implementation.

A model similar to that of Simon often referenced by defense researchers in decision making is the OODA Loop (shown in Figure 2.2), an unpublished model by Boyd describing the actions of *Observe–Orient–Decide–Act* (Tweedale *et al*. 2008). Boyd's model was based on observation of the superiority of US pilots in dogfights during wartime even when they were outgunned. During the *Observe* phase the pilot could see their adversary earlier and better, allowing them to *orient* to the threat. As decision makers, US pilots were able to *decide* using their training to assess available information, and then *act* more quickly than their adversary. Boyd presented his research in two seminal lectures on 'patterns of conflict' and 'a discourse on winning and losing' that have significantly influenced military tactics and decision making (Tweedale *et al*. 2008).

Simon's model of Bounded Rationality and his view of decision making as a process was descriptive of managerial decision making (Simon 1977, 1997), and, thus, became the model on which decision support systems were based. In Simon's model, the decision maker does not require complete knowledge of events, and the goal is satisfactory decisions which may, in fact, be sub-optimal. Simon referred to these decisions as 'satisficing' as compared with 'maximizing' (Byron 2004). Satisficing allows that it is rational to choose any action that leads to a satisfactory outcome, while maximizing requires rational agents to assign a utility or preference to all possible outcomes of every alternative action and select the one with the maximum utility. Decision making involves possible contradictory criteria as well as multiple criteria in Simon's view.

Research has shown that humans will use strategies such as rationalization, reasoning by analogy, heuristics, trial and error, local adaptations, avoidance of loss, and control of risk to avoid multi-criteria choices (Pomerol and Adam 2008). Such sub-optimal strategies can lead to cognitive bias, with several well-known errors in decision making: anchoring, status quo, confirming evidence, framing, estimating and forecasting, and

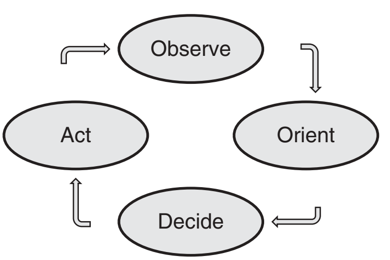

Figure 2.2 Boyd's OODA loop.

sunk cost (Hammond *et al*. 1999). Anchoring is the disproportionate weight given to some information, such as that received first, and which forms the basis for subsequent comparison. Status quo is an effort to maintain the current state. Confirming evidence refers to seeking information that supports the decision while avoiding information that does not. The framing phenomenon is a selection of an alternative depending on the way that it is presented, such as presenting the decision in terms of the amount of gain or the amount of loss. Estimating and forecasting occurs when decision makers overestimate their ability to perform these two tasks, particularly seen in estimating uncertain events. Finally, sunk cost occurs when decision makers make choices to justify past decisions, even when such a strategy can be shown to be a poor decision for the future.

IDSS help overcome human cognitive limitations and biases by providing a rational basis for comparison of alternatives. We first discuss the basics of traditional DSS before turning to the integration of AI techniques to form IDSS.

2.3 Decision support systems

DSS refer to a broad range of interactive computer systems that assist decision makers to utilize data, models and knowledge to solve semi-structured, ill-structured, or unstructured problems (Sprague and Watson 1996). The decision maker is part of the system, so DSS include capabilities to allow the decision maker to do one or more of input selections, query the system, drill-down for explanations, examine output, and generally interact with the computing device. Since most DSS are developed to solve a specific problem or general class of problems, there are many types of DSS specialized for different types of users and problems. DSS can also be designed for one or multiple decision makers, and can be used to support decisions that range from managerial to creative problem solving. Various terms have been attached to DSS including Expert Systems, Group DSS, Collaborative DSS, Adaptive DSS, Clinical DSS, Executive Support Systems, Intelligent DSS and so on, that attempt to capture either the purpose of the DSS or the user(s) targeted by the DSS (Keen and Scott-Morton 1978). Single-user DSS are generally based on Simon's (1997) process of decision making and have components for Input (Intelligence), Processing (Design), and Output (Choice). Group or Collaborative DSS are in the early development stage since theories of collaborative decision making among humans are still emerging. Similarly, DSS to support innovation and idea generation are emerging along with theories of the human creative process.

A schematic of the basic structure of a decision support system is shown in Figure 2.3 and includes the decision maker as part of the system. Input includes a database, knowledge base and model base. The database contains data relevant to the decision problem while the knowledge base may include, for example, guidance in selecting alternatives. The model base holds formal models, algorithms and methodologies for developing outcomes. Processing involves using decision models to simulate or examine different states and find the best solution under the constraints. Feedback from processing can provide additional inputs that may be updated in real-time to improve problem solution. Output may generate forecasts and explanations to justify recommendations and offer advice. Results are provided to the decision maker who may interact with the system to provide additional information or query the results.

More recently, the term decision support has been broadened to include other decision making aids such as knowledge management systems, business intelligence, and analytics,

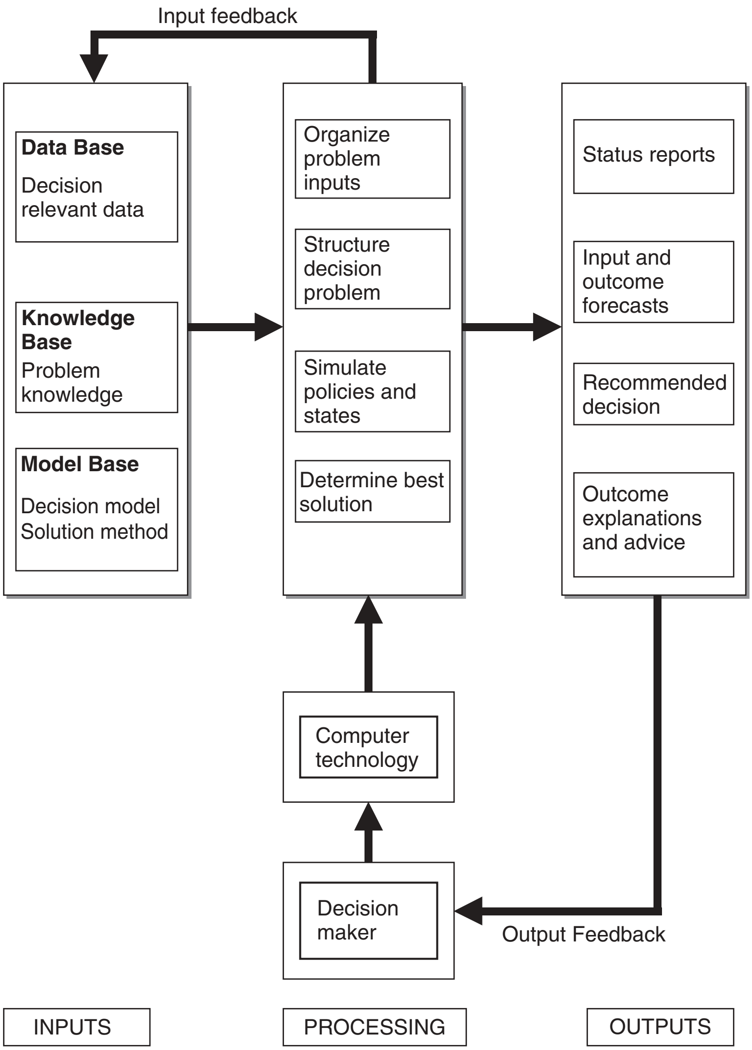

Figure 2.3 Structure of a decision support system (Phillips-Wren *et al*. 2009).

that may or may not include interaction with the decision maker. AI features are often used to expand the capabilities of these decision aids to, for example, aggregate widely dispersed data and draw observations from large distributed datasets called 'big data.' Such systems can include features such as personalization for decision maker preferences or even emulate human decision making. They offer powerful new tools to solve highly complex problems and are emerging trends for the future.

2.4 Intelligent decision support systems

IDSS utilize AI techniques to enhance and improve support for the decision maker. AI tools such as fuzzy logic, case-based reasoning, evolutionary computing, artificial neural

networks (ANN), and intelligent agents, when combined with DSS, provide powerful aids in solving difficult applied problems that are often real-time, involve large amounts of distributed data, and benefit from complex reasoning.

One way of looking at intelligence is that it is primarily concerned with rational action, so that an intelligent system would take the best possible action in a situation (Russell and Norvig 1995). In order to do so, IDSS would be DSS that exhibit some abilities indicative of 'intelligent behavior' (Turban and Aronson 1998) such as:

- learning from experience;
- making sense out of ambiguity or contradiction;
- responding appropriately and timely to a new situation;
- using reasoning to solve problems and inferring in rational ways;
- dealing with perplexing situations;
- applying knowledge to understand or change the environment;
- recognizing the relative importance of various factors in a decision.

IDSS are emerging as useful for practical and important applications as seen in Table 2.1 and employ a variety of AI techniques. Applications range from healthcare support to business decisions, all with the capability to improve human decision making. We will discuss fundamentals of the more common AI techniques and their contribution to decision making in the sections that follow.

2.4.1 Artificial neural networks for intelligent decision support

ANN (or neural networks, NN, for short) are composed of a collection of highly interconnected processing units called neurons that work together to solve a problem. NN were inspired by the way that the brain processes information and are generally composed of layers of neurons as shown in Figure 2.4. The advantage of NN is their ability to represent any bounded continuous function to any arbitrarily small approximation error (SAS 2012).

The basic unit of a NN is the neuron or node. Each neuron receives an input x_i as a stimulus, with an associated weight w_i expressing the relative importance of x_i, from another neuron or an external source. Weights may be positive or negative, i.e., excitatory or inhibitory. The neuron computes the weighted sum of all inputs to the neuron, i.e., for the jth neuron,

$$Y_j = \sum_i x_i w_{ij}$$

where Y_j represents the activation level of the neuron (Pedrycz *et al*. 2008). Information can be modified and transferred from the neuron using several different types of transformation functions which may be nonlinear. The weights w_i are critical to learning in the NN since they are adjusted as new data become available. Typically the NN are exposed iteratively to a training set of data that includes actual associated outputs to develop a set of weights until the outputs from the NN match the actual outputs to a desired level of accuracy. The NN can then be used to predict future states with a set of inputs, or 'learn'

Table 2.1 Recent reported applications of intelligent decision support systems.

Author	AI tools	Description of IDSS
Kaszuba and Kostek (2012)	Fuzzy logic	Hand gesture classification for testing and monitoring patients with neurological conditions such as Parkinson disease
Lam *et al*. (2012)	Case-based algorithm Genetic algorithm	Warehouse cross-border delivery activities, such as palletization of the delivery goods according to regulation requirements
Lao *et al*. (2012)	Fuzzy rule-based reasoning Case-based reasoning	Quality control of food inventories in the warehouse
Kung *et al*. (2012)	Intelligent agents Neural network (back-propagation)	Predict and decide the debris-flow occurrence in Taiwan in disasters
Saed *et al*. (2012)	PSO algorithm	Optimal design that satisfies quality requirements based on level of redundancy and trade-off between reliability and performance
Lei and Ghorbani (2012)	Neural networks	Detection of fraud and network intrusion
Santos *et al*. (2011)	Intelligent agents incorporating affective characteristics (such as personality, emotion, and mood)	Decision aid to help improve the negotiation process
Stathopoulou and Tsihrintzis (2011)	Neural network	Rapid and successful detection of a face in an image for automated face recognition system
Buditjahjanto and Miyauchi (2011)	Genetic algorithm Clustering	Tradeoff between fuel cost (economic) and emission problems to achieve optimal decisions
Kamami *et al*. (2011)	Fuzzy logic	Selection of sustainable wastewater treatment technologies
Dolsak and Novak (2011)	Expert system	Structural design analysis using a finite element method

PSO, particle swarm optimization.

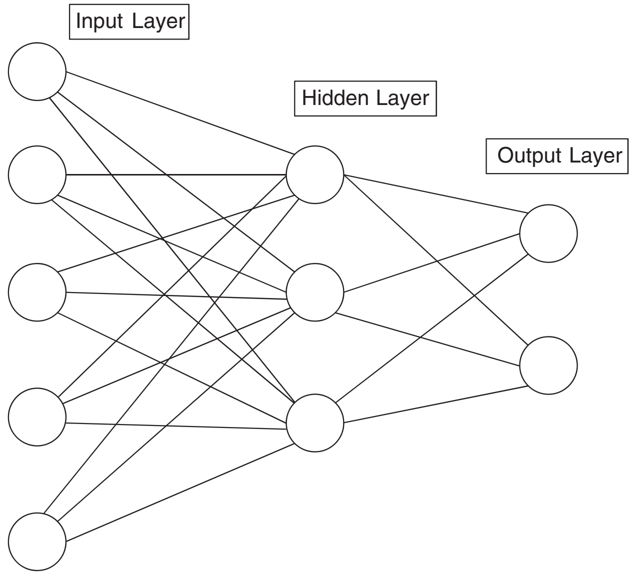

Figure 2.4 A neural network with a hidden layer.

to readjust weights as additional input/output datasets are provided. This feature enables NN to find patterns and generalize based on previous cases, and to classify based on observation, much as humans use empirical observations to suggest outcomes from past history or observed behavior (Turban and Aronson 1998).

Two generic topologies of NN are feedforward and recurrent, or feedbackward, networks. As the name suggests, in feedforward NN signals flow forward from inputs to outputs, and possibly though multiple hidden layers. For example, Figure 2.4 shows four inputs that flow through a hidden layer consisting of three nodes to the output layer. Feedforward NN have been most often utilized in decision problems since the flow of information is consistent with the process of decision making (Phillips-Wren 2012). Feedbackward NN can have signals traveling in both directions.

NN are different from sequential, logic-based approaches that assume a specific type of relationship between inputs and output. Their fundamental advantage is that they can represent nonlinearity naturally as part of the process of fitting the weights. NN offer decision support for important problems such as fraud detection that are nearly impossible to model with logic-based approaches (Lei and Ghorbani 2012). On the other hand, NN are not suitable for operations such as data processing.

NN ‘learn’ the underlying function described by the data using one of three strategies with ‘training data’: unsupervised, supervised, or reinforcement learning. Unsupervised learning occurs when the NN are given only inputs and no corresponding outputs. The goal is to identify underlying structure in the data. In supervised learning, the NN are given inputs and corresponding outputs to adjust weights on the various inputs so that outputs produced by the NN are within the desired approximation error to the sample outputs (targets). The identification of weights on the inputs permits the NN to be used for prediction with a new set of inputs. However, in many practical situations, detailed input–output n-tuples are not available, and there may only be a limited or even rare

number of outputs after a long number of inputs. Reinforcement learning is used to deal with this situation and provide some feedback to the NN to evaluate whether the weights were chosen correctly. When training NN for a decision problem, over-fitting (or too closely matching the training data) needs to be avoided so that the NN can be generalized and used for predictive purposes.

NN are sometimes referred to a ‘black boxes’ meaning that the interpretation of the model is difficult for the decision maker. Since the computation is distributed over several or many nodes and hidden layers, decision rules and relationships between variables are not easily discernible from the NN.

2.4.2 Fuzzy logic for intelligent decision support

Fuzzy logic extends decision support by permitting a representation of inputs or variables in the decision problem the way that humans reason about them. Decision makers often encounter problems in which inputs are imprecise or uncertain. For example, the weather may be sunny, partly sunny, mostly cloudy, or cloudy. By comparison, Boolean logic is a system of symbolic logic that governs logical functions on a computer and is based on a binary system of 0 (completely false) and 1 (completely true). Fuzzy logic enables a representation of uncertainty by allowing an input to have a range of values between 0 (completely false) and 1 (completely true).

Fuzzy logic can be advantageous in IDSS by providing (Turban and Aronson 1998):

- flexibility to make allowances for the unexpected;
- options for intuition such as ‘likely’ or ‘very good’;
- ability to imagine what-if scenarios;
- low risk for incorrect choices since there is a range of values to choose from;
- modeling approaches for problems with uncertainty that are difficult to represent in mathematical models.

Fuzzy logic is more flexible in representation, so that the decision maker has a range of choices and is free to estimate values of inputs. There is no inherent structure in fuzzy logic, so nonlinear relationships can be encapsulated naturally without prior planning. As new information becomes available, values can be refined and modified easily, giving the decision maker a natural way to deal with uncertainty. Fuzzy logic provides a way to represent rule-based behaviors, such as knowledge from an expert, so that expertise can be captured and provided to the decision maker at the appropriate time (Phillips-Wren 2012). Fuzzy logic can also be combined with NN so that the interpretation of decision variables is more apparent. For example, input variables might be described with three values such as a maximum, minimum, and most likely value. These are natural language descriptions that enhance the ability of a decision maker to communicate domain knowledge to a model and interpret output.

Fuzzy logic NN are a category of decision models that can provide human understandable meaning to the multilayer feedforward NN (Lin and Lee 1991). Aggregative neurons accomplish AN-OR logic while referential neurons support predicate-based processing such as *less than*, *greater than*, or *similar* expressed in the language of fuzzy logic. Fuzzy logic NN help address some of the shortcomings of NN regarding transparency to enhance decision making.

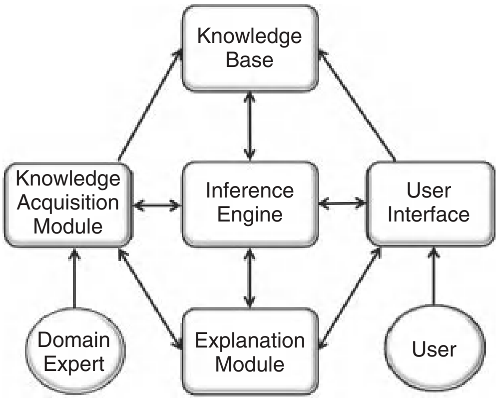

Figure 2.5 Components of an expert system (Ahmad A, Basir O and Hassanein K (2008) Intelligent expert systems approach to layout decision analysis and design under uncertainty. In *Intelligent Decision Making: An AI-Based Approach* (eds Phillips-Wren G, Ichalkaranje N and Jain L). Springer, Berlin, pp. 321–364. With kind permission from Springer Science and Business Media).

2.4.3 Expert systems for intelligent decision support

An expert system (ES) is a computer system that attempts to solve problems that would normally be solved by a human expert (Liao 2004). Often the term ES is used to describe a system that embeds the intelligence of one or more identified human experts. The system designer needs to study how the human expert makes the decision, and then embed that knowledge into the computer system.

Components of an ES are shown in Figure 2.5. It shows a domain expert who provides knowledge to the Knowledge Acquisition Module. That knowledge is encoded in the Knowledge Base, usually as part of the development process. The user or decision maker enters the system through an interface. The user may then directly access the Knowledge Base for past cases, or the Inference Engine to infer from past cases to a new case. The user may 'drill down' for an explanation of the inference from the Explanation Module. The ES thus serves to capture, collect and infer knowledge from a domain expert and pass that expertise to a decision maker.

2.4.4 Evolutionary computing for intelligent decision support

Evolutionary computing draws inspiration from natural evolution where individuals in a population evolve to increase their survival rate by increasing their fitness with respect to a goal (Eiben and Smith 2003). AI techniques attempt to mimic these qualities of adapting to the environment by simulating emergence, survival and refinement of a population of individuals. Genetic algorithms (GA) are among the most utilized for decision problems. After the population is initialized, succeeding generations interact, communicate, and influence each other in order to better match the environment.

A generalized flowchart for a genetic algorithm is shown in Figure 2.6. A finite population is randomly initialized at time $t = 0$, and a goal is stated. Each individual is evaluated with respect to the goal and characterized with a fitness value that represents the suitability of the individual for the environment. The higher the fitness value, the

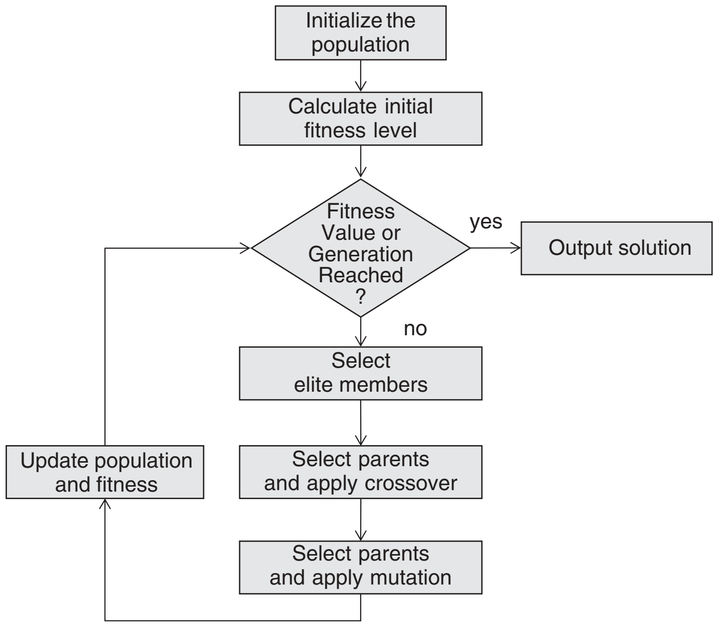

Figure 2.6 Generalized flowchart for a genetic algorithm.

more likely the individual is to survive and become a parent for succeeding generations. Weak individuals may also become parents. The population continues to evolve until the stopping criteria are met or the number of generations is accomplished. Over time the population becomes more fit and focused. Two techniques are used to refine the population: crossover and mutation. Crossover requires selected individuals to swap pieces of their code to produce offspring. Mutation causes small alterations in some components of an individual's code. Evolutionary computing provides insight into decision strategies for solving multicriteria problems (Fonseca and Fleming 1998).

2.4.5 Intelligent agents for intelligent decision support

Of the various AI techniques, intelligent agents (IA, a class that includes agents) have had the broadest applicability to decision problems (Jain *et al*. 2008). By way of a definition, an agent is an entity within a system that 'is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objective' (Wooldridge 2002, p. 23). This intrinsic capability of autonomous action is the human-like characteristic of acting (or deciding) on the basis of context rather than the prescriptive logic embedded in if–then computer programs.

The literature makes a distinction between an agent and an intelligent agent with the concepts of weak and strong agencies (Woolridge and Jennings 1995). Weak agency includes capabilities such as autonomy, reactivity, adaptiveness, proactivity and social ability. Strong agency adds to these abilities more advanced characteristics such as communication, persistence, mobility, rationality and learning. Autonomy is the ability to make decisions independently of others. Agents that display reactivity and adaptiveness perceive the environment and respond as the environment changes. Proactiveness means that the IA can initiate action with a specific directive to meet a goal. Social ability

allows agents to interact with others. Communication with other agents is displayed by agent social ability and leads to traits such as cooperation and negotiation. Persistence means that the agent maintains state over a long period of time, and mobility permits the agent to travel throughout the system to perform tasks or search for new knowledge. Rationality is the ability to make decisions based on the environment, and learning means that the IA can change their responses based on past interactions or situations. Human terms such as knowledge, intention and beliefs are used to describe IA, indicating the complex behaviors that are possible.

A powerful method of supporting decision tasks for complex situations is to create multi-agent systems (MAS) comprised of agent teams that do not have complete information of the environment or of other agents (Jain *et al*. 2008). Each intelligent agent can be invested with particular characteristics, and individual agents can team with other agents to act on behalf of the decision maker or other agents with different goals and objectives (Padgham and Winikoff 2004). Agents may need to interact, coordinate, negotiate, learn and even trust other agents to achieve their goals (Huhns and Singh 1998). Decision makers can receive recommendations from the MAS or allow agents to act without specific authorization from a human. MAS are an active area of research in complex applications and involve issues around representation of the decision problem, team size, and reconfiguring teams during the decision problem solution.

Dynamic, uncertain environments are particularly well-suited to MAS. Agent teams can perceive the environment, adapt to changes, share knowledge among team members, and arrive at a negotiated action plan. Agents can learn using supervised, unsupervised or reward-based learning (Russell and Norvig 1995). Similar to NN, in supervised learning the system receives correct outputs to guide the system. In unsupervised learning no outputs are provided so that the system needs to identify structure in the problem. In reward-based learning the MAS receive positive or negative reinforcement which guides the systems.

Basic notions of agent technology are (Phillips-Wren 2012):

- Communication – needed to convey the intentions of the IA. Communication can be direct or indirect. Direct communication can occur with message passing or shared variables. Indirect communication can occur inferentially, for example, by an agent observing the actions of other agents.
- Coordination – needed to organize IA, their resources and tasks. Coordination is used to resolve conflicts and improve agent performance.
- Teaming agreements – needed to codify the behaviors such as communication and coordination of IA to interact and act as a team to pursue a common objective.
- Human-centric agents – needed for rich interaction between the computer system and a human decision maker. For example, sub-tasks that require human judgment can be presented to the human decision maker as appropriate. Recent research has focused on this attribute in MAS.
- Learning – needed to make IA more human-like and facilitate the interaction between MAS and human decision makers.

Recent research in MAS for complex decision problems has focused on a framework called Belief–Desire–Intention (Sioutis *et al*. 2003). An agent's beliefs are its understanding of the external world; its desires are the goals to achieve; and, its intentions

are the actions that are needed to satisfy its desires. In other words, what does the agent want and how does it intend to satisfy that desire knowing what it believes is true? An external world view can be provided for situation awareness or context for decision making. Context evolves with the decision problem and can be highly domain-dependent. Representing and including context within IDSS is an emerging area of research.

2.5 Evaluating intelligent decision support systems

2.5.1 Determining evaluation criteria

Evaluating IDSS, or any DSS for that matter, is critical to understanding both benefits and opportunities for enhancing the system. We briefly present the primary research on this important, and often neglected, area. A study of the literature indicates that researchers often report a single outcome criterion for evaluating the success of a DSS such as improved efficiency or increased speed in making the decision (Phillips-Wren *et al*. 2009). These outcomes can often be associated with a tangible benefit such as reduced cost or increased revenue. However, a closer examination reveals that process outcomes are also cited as a benefit of DSS/IDSS. As defined by Simon (1977), the process of decision making consists of intelligence, design, choice and implementation. An IDSS could, for example, perform database searches autonomously as the system perceives the needs of the decision maker or assist the user in selecting variables in the design phase. Even in cases in which the overall outcome is not changed, the decision maker may have a better understanding of the decision problem by using the IDSS. Therefore, evaluation of DSS/IDSS should be multi-criteria, and insights from the evaluation can guide design and improvement of the system.

Multi-criteria evaluation of information systems has been examined in the literature (DeLone and McLean 1992, 2003), with some applied directly to DSS (Sanders and Courtney 1985). Chandler (1982) noted that information systems create a relationship between the system and its users, and he developed tradeoffs between goals and performance using a multiple goal programming approach. Adelman (1992) applied these ideas to DSS and expert systems with a multi-criteria approach consisting of subjective, technical, and empirical methods. Subjective criteria included user and sponsor perspectives; technical criteria included analytical methods and correctness of the analysis; and, empirical criteria included comparisons of performance with and without the system.

Further, Turban and Aronson (1998) evaluated information systems with two major classes of performance measurement: effectiveness and efficiency. Checkland (2000) used three criteria: general systems principles to evaluate how well the outputs contribute to the achievement of goals; efficiency to measure how well inputs and resources are processed to achieve outputs; and efficacy to evaluate how well the system produces the expected outputs. Efficacy is a values-based perspective.

Turban and Aronson (1998) pointed out that DSS affect both the process of, and outcome from, decision making. Forgionne (1999), later with Phillips-Wren *et al*. (2008), noted that published DSS applications from a wide variety of sources assessed the system using either the process of, or outcome from, decision making. Process criteria improved the way the decision was made and were often measured in qualitative terms such as increased efficiency, more systematic process, better understanding of the problem, ability to generalize results, and speed of reaching a decision. Outcome criteria were assessed

in quantifiable terms such as increased profit, decreased cost, accuracy of prediction, and success in predicting failure. Thus, multi-criteria assessment of IDSS is justified in both theory and practice.

2.5.2 Multi-criteria model for IDSS assessment

The literature proposes that IDSS should be assessed on the basis of both the process of, and outcome from, decision making. One way to develop a quantitative model for the 'decision value' of an IDSS (Phillips-Wren *et al*. 2009) is to use the analytic hierarchy process (AHP; Saaty 1977, 1986). An advantage of AHP is that the individual contributions of various subcomponents can be determined. A stochastic enhancement of AHP is possible and allows the statistical significance of the contributions to be determined (Phillips-Wren *et al*. 2004).

AHP provides a methodology for comparing alternatives by structuring criteria into a hierarchy relevant to the system or to the decision problem. The AHP hierarchy is designed to break down relevant criteria into levels from higher level criteria to lower level criteria much as a reporting diagram for an organization. The evaluator only needs to supply pairwise comparisons of alternatives at the lowest level, and the AHP computes all intermediate comparisons and combines them into a decision value that compares alternatives at the top level. Criteria can be weighted, if desired, and an eigenvalue solution is used to reconcile initial judgments in cases where there are more than two alternatives. AHP has been used extensively for decision problems (Saaty and Vargas 1994), and it can be applied in general to evaluate IDSS (Phillips-Wren *et al*. 2009).

A proposed AHP evaluation of a generalized IDSS is shown in Figure 2.7. The hierarchy is designed to connect the computational method used to solve the decision problem with the decision value of an IDSS using the various methods. The

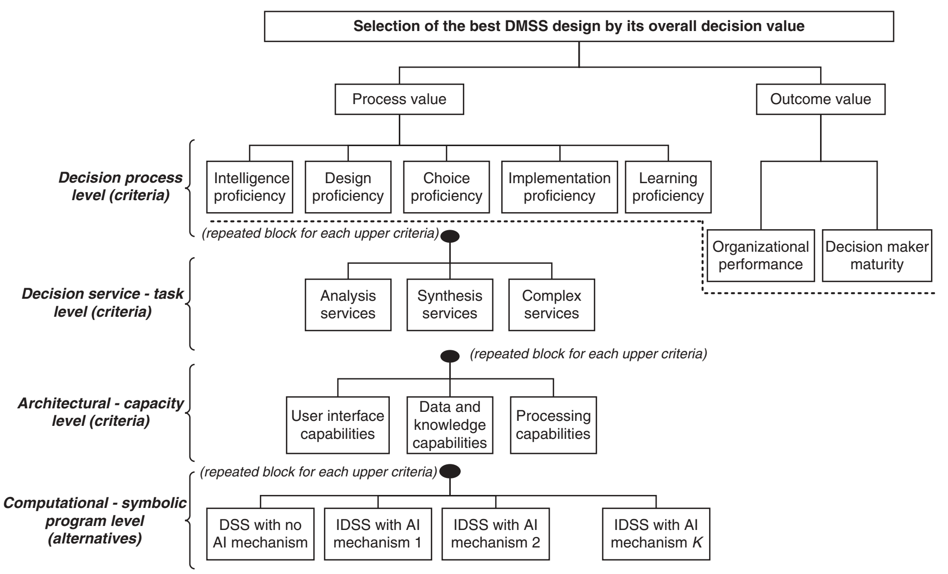

Figure 2.7 Multiple criteria model to evaluate an IDSS (Phillips-Wren *et al*. 2009).

computational level reports up to the architecture of the IDSS (user interface, data and knowledge capabilities, and processing) which, in turn, reports up to the decisional tasks (analysis, synthesis, complex). Decisional tasks report up to the process level. Outcomes to be evaluated are provided by the decision maker to produce an outcome level. The combination of process and outcome is combined to yield the decision value of the alternatives that were introduced at the lowest level. Multi-criteria evaluation of IDSS can provide insight into the design of the system and highlight areas needing improvement, resulting in more robust support for decision making. For more details and an applied example, see Phillips-Wren *et al*. (2009).

2.6 Summary and future trends

Emerging research in the application of AI to decision support is focused in three areas: smart adaptive systems that can modify themselves to solve a complex problem or the preferences of an individual user; interfaces such as virtual humans that can increase the effective interaction between humans and machines; and, techniques applicable to extremely difficult applied problems such as big data.

Advances in computing power and storage has led to the capture of vast amounts of data and information that can be used as the foundation for decision making in such fields as financial services, point-of-sale transactions, user profiles via the Internet, and healthcare. Such problems tend to be complex, imprecise, noisy, dynamic and uncertain. Adaptation is essential to solving such problems by utilizing the most effective AI technique or combination of techniques (hybridization) for the domain. One way to define smart adaptive systems utilizing intelligent techniques is to separate the problems into three increasingly challenging levels: adaption to a changing environment; adaption to a similar setting without direct porting to it; and, adaptation to a new or unknown situation (Gabrys *et al*. 2005). Changing environments are characterized by the speed of change or time for the decision and the amount of data available; for example, real-time decision support is needed for financial decisions such as the stock market based on algorithmic trading. A similar setting is an ability to generalize from previous cases; for example, identifying possible fraudulent activity on a credit card based on past history and informing the user immediately. The most difficult problems are completely new or unknown situations. Humans use a variety of approaches for such problems, including context, past experience, and the advice of experts. Systems are emerging for practical applications in specific domains but have yet to be generalized for classes of problems.

Intelligent interfaces is an evolving area that includes human–computer interaction. One way to improve the way that humans and computers communicate is through virtual humans (Kasap and Magnenat-Thalmann 2007). Three important aspects of interaction are autonomy (such as action and movement), interaction (such as facial expressions) and personification (such as personality and emotion) that consider the mental state of the virtual human in its interaction with an actual human. A verbal interface with a machine is being developed in the field of natural language processing. For example, a 'commonsense' knowledge base and 'commonsense' inference technique was developed to engage humans in storytelling with a machine (Chi and Lieberman 2011). Another example is the Semantic Web, envisioned as way that intelligent agents can roam Web pages to carry out sophisticated tasks for humans by perceiving meaning from human–machine interaction (Berners-Lee *et al*. 2001). Intelligent multimedia are being developed to address

challenges in interaction in which multimedia modalities are used (Virvou and Tsihrintzis 2012). For example, decisions related to medical diagnosis based on imagery such as cancer screening can be assisted with intelligent techniques that interpret images and act as second decision makers. Other types of multimedia in which data and knowledge are embedded include video, graphics, text, audio, speech, digital signals, voice, spatial representations, and imagery. The exponential expansion in such data sources, along with the ability to store and retrieve them, has necessitated automated intelligent processing to aid human decision makers. This active research field includes new methodologies and approaches to solving these complex problems.

As the amount of data has exploded over distributed systems and databases linked by the Internet, new terms have been created to describe the size and scope of the problems that can be addressed. ‘Big Data’ is a recent term in the practitioner literature used to describe the exponential growth in volume, availability and use of information (Beyer 2011). One definition of ‘big data’ is ‘large, diverse, complex, longitudinal, and/or distributed data sets generated from instruments, sensors, Internet transactions, email, video, click streams, and/or all other digital sources available today and in the future’ (NSF 2012). Research issues that arise from ‘big data’ include how to analyze the huge *volume* of data; how to utilize the *variety* of information (i.e. dimensionality such as tabular data, documents, email, video, imagery, audio, stock ticker data, financial data, transactional data, etc.) as a basis for decision making; and, how to store and process data arriving at high *velocity* for enhanced decision making (Beyer 2011). Manyika *et al*. (2011) pointed out that ‘sophisticated analytics [applied to big data] can substantially improve decision-making’ by, for example, collecting more accurate performance information, providing detailed information on underlying contributors to performance, conducting controlled experiments for management decisions, forecasting and nowcasting to adjust business levers, segmenting customers narrowly to tailor products or services more precisely, and using embedded sensors in products for innovative services such as proactive maintenance.

Future research in intelligent decision technologies presents exciting opportunities and challenges. Opportunities for improved decision making are significant, particularly in complex problems that are beyond human capabilities to perceive relationships between variables, are dynamic in nature, encapsulate rare events that are difficult to perceive, or require huge amounts of data to uncover the underlying structure of the problem. The challenge for broad applicability of IDSS is to design systems that have a clear return on investment, can effectively interact with humans, can gain human trust in the system, and are able to solve impactful applied problems. Such systems can lead to a new wave of innovation and vastly improved human decision making.

Acknowledgment

The author would like to thank the many researchers whose work has provided the foundation for this review.

References

Adelman L (1992) *Evaluating Decision Support and Expert Systems*. John Wiley & Sons, Ltd, Hoboken, NJ.

Adolphs R, Tranel D, Bechara A, Damasio H and Damasio AR (1996) Neuropsychological approaches to reasoning and decision making. In *Neurobiology of Decision-Making* (eds Damasio AR, Damasio H and Christen Y). Springer, Berlin, pp. 157–179.

Ahmad A, Basir O and Hassanein K (2008) Intelligent expert systems approach to layout decision analysis and design under uncertainty. In *Intelligent Decision Making: An AI-Based Approach* (eds Phillips-Wren G, Ichalkaranje N and Jain L). Springer, Berlin, pp. 321–364.

Berners-Lee T, Hendler J and Lassila O (2001) The semantic web. http://www.scientificamerican.com/article.cfm?id=the-semantic-web. Scientific American Magazine, Accessed on March 22, 2012.

Beyer M (2011) Gartner says solving 'big data' challenge involves more than just managing volumes of data. http://www.gartner.com/it/page.jsp?id=1731916. Gartner, Accessed on April 12, 2012.

Buditjahjanto IGPA and Miyauchi H (2011) An intelligent decision support based on a subtractive clustering and fuzzy inference system for multiobjective optimization problem in serious game. *International Journal of Information Technology and Decision Making* **10**(5), 793–810.

Byron M (2004) *Satisficing and Maximizing: Moral Theorists on Practical Reason*. Cambridge University Press, New York.

Chandler J (1982) A multiple criteria approach for evaluating information systems. *Management Information Systems Quarterly* **6**(1), 61–74.

Checkland P (2000) *Systems Thinking, Systems Practice*. John Wiley & Sons, Ltd, Chichester.

Chi PY and Lieberman H (2011) Intelligent assistance for conversational storytelling using story patterns. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*. ACM, New York, pp. 217–226.

Damasio A (1994) *Descartes' Error*. Putnam's Sons, New York.

DeLone W and McLean E (1992) Information systems success: The quest for the dependent variable. *Information Systems Research* **3**(1), 60–95.

DeLone W and McLean E (2003) The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems* **19**(4), 9–30.

Dolsak B and Novak M (2011) Intelligent decision support for structural design analysis. *Advanced Engineering Informatics* **25**(2), 330–340.

Eiben A and Smith J (2003) *Introduction to Evolutionary Computing*. Springer-Verlag, New York.

Fonseca CM and Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms-part I: A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics–Part A* **28**(1), 26–37.

Forgionne G (1999) An AHP model of DSS effectiveness. *European Journal of Information Systems* **8**, 95–106.

Gabrys B, Leiviska K and Strackeljan J (2005) *Do Smart Adaptive Systems Exist?* Springer-Verlag, Berlin.

Halpern J, Blume L and Easley D (2012) Redoing the foundations of decision theory. http://www.cis.upenn.edu/timedc/may05/joseph.pdf. Gartner, Accessed on February 10, 2012.

Hammond J, Keeney R and Raiffa H (1999) The hidden traps in decision making. *Harvard Business Review* **84**(1), 118–126.

Huhns M and Singh M (1998) *Readings in Agents*. Morgan Kaufmann Publishers, San Francisco, CA.

Jain L, Tan S and Lim C (2008) An introduction to computational intelligence paradigms. In *Computational Intelligence Paradigms* (eds Jain L, Sato-Ilic M, Virvou M, Tsihrintzis G, Balas V and Abeynayake C), vol. 137 of *Studies in Computational Intelligence*. Springer, Berlin, pp. 1–23.

Kamami M, Ndegwa G and Home P (2011) Fuzzy based decision support method for selection of sustainable wastewater treatment technologies. *International Journal of Agricultural and Biological Engineering* **4**(1), 41–51.

Karni E (2005) Savage's subjective expected utility model. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.541.pdf. Gartner, Accessed on January 12, 2012.

Kasap Z and Magnenat-Thalmann N (2007) Intelligent virtual humans with autonomy and personality: State-of-the-art. *Intelligent Decision Technologies* **1**(1–2), 3–15.

Kaszuba K and Kostek B (2012) A new approach for automatic assessment of a neurological condition employing hand gesture classification. *Intelligent Decision Technologies* **6**(2), 171–176.

Keen P and Scott-Morton M (1978) *Decision Support Systems: An Organizational Perspective*. Addison-Wesley, Reading, MA.

Klein GA (1993) A recognition-primed decision (RPD) model of rapid decision making. In *Decision Making in Action: Models and Methods* (eds Klein GA, Orasanu J, Calderwood R and Zsambok CE). Ablex Publishing Corporation, Norwood, NJ, pp. 138–147.

Kung H, Chen C and Ku H (2012) Designing intelligent disaster prediction models and systems for debris-flow disasters in Taiwan. *Expert Systems with Applications* **39**(5), 5838–5856.

Lam CHY, Choy KL, Ho GTS and Chung SH (2012) A hybrid case-GA-based decision support model for warehouse operation in fulfilling cross-border orders. *Expert Systems with Applications* **39**(8), 7015–7028.

Lao SI, Choy KL, Ho GTS, Yam RCM, Tsim YC and Poon TC (2012) Achieving quality assurance functionality in the food industry using a hybrid case-based reasoning and fuzzy logic approach. *Expert Systems with Applications* **39**(5), 5251–5261.

Lei J and Ghorbani A (2012) Improved competitive learning neural networks for network intrusion and fraud detection. *Neurocomputing* **75**(1), 135–145.

Liao S (2004) Expert system methodologies and applications – A decade review from 1995 to 2004. *Expert Systems with Applications* **28**(1), 93–103.

Lin C and Lee C (1991) Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers* **40**(12), 1320–1336.

Manyika J, Chui M, Brown B, Bughin J, Dobbs R, Roxburgh C and Byers AH (2011) Big data: The next frontier for innovation, competition, and productivity. http://www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation. McKinsey Global Institute, Accessed on May 2, 2012.

NSF (2012) Core techniques and technologies for advancing big data science & engineering (BIGDATA). http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=504767&WT.mc_id=USNSF_39&WT.mc_ev=click. Accessed on May 12, 2012.

Padgham L and Winikoff M (2004) *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley & Sons, Ltd, Hoboken, NJ.

Pedrycz W, Ichalkaranje N, Phillips-Wren G and Jain L (2008) Introduction to computational intelligence for decision making. In *Intelligent Decision Making: An AI-Based Approach* (eds Phillips-Wren G, Ichalkaranje N and Jain L). Springer, Berlin, pp. 79–96.

Phillips-Wren G (2012) AI tools in decision making support systems: A review. *International Journal on Artificial Intelligence Tools*, **21**(2), April, 13 pages.

Phillips-Wren G, Hahn E and Forgionne G (2004) A multiple criteria framework for the evaluation of decision support systems. *Omega* **32**(4), 323–332.

Phillips-Wren G, Mora M and Forgionne G (2008) Evaluation of decision making support systems. In *Encyclopedia of Decision Making and Decision Support Technologies* (eds Adam F and Humphreys P). IGI Publishing, Hershey, PA, pp. 320–328.

Phillips-Wren G, Mora M, Forgionne G and Gupta J (2009) An integrative evaluation framework for intelligent decision support systems. *European Journal of Operational Research* **195**(3), 642–652.

Pomerol JC (1997) Artificial intelligence and human decision making. *European Journal of Operational Research* **99**, 3–25.

Pomerol JC and Adam F (2008) Understanding human decision making – A fundamental step towards effective intelligent decision support. In *Intelligent Decision Making: An AI-Based Approach* (eds Phillips-Wren G, Ichalkaranje N and Jain L). Springer, Berlin, pp. 3–40.

Russell S and Norvig P (1995) *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.

Saaty T (1977) A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology* **15**(3), 234–281.

Saaty T (1986) How to make a decision: The analytic hierarchy process. *Interfaces* **24**(6), 19–43.

Saaty T and Vargas L (1994) *Decision Making in Economic, Political, Social and Technological Environments with the Analytic Hierarchy Process*. RWS Publications, Pittsburgh, PA.

Saed A, Kadir W, Hamza H and Yousif A (2012) An automated support for evaluating alternative design decisions. *Journal of Theoretical and Applied Information Technology* **36**(2), 234–246.

Sanders G and Courtney J (1985) A field study of organizational factors influencing DSS success. *Management Information Systems Quarterly* **9**(1), 77–89.

Santos R, Marreiros G, Ramos C, Neves J and Bulas-Cruz J (2011) Personality, emotion, and mood in agent-based group decision making. *IEEE Intelligent Systems* **26**(6), 58–66.

SAS (2012) Neural networks. http://www.sas.com/technologies/analytics/datamining/miner/neuralnet/index.html. Accessed on May 4, 2012.

Savage LJ (1954) *The Foundations of Statistics*, 2nd edn. Dover Publications, New York.

Simon HA (1955) A behavioural model of rational choice. *Quarterly Journal of Economics* **69**, 99–118.

Simon HA (1977) *The New Science of Management Decision*, 3rd edn. Prentice-Hall, Englewood Cliffs, NJ.

Simon HA (1997) *Administrative Behavior*, 4th edn. The Free Press, New York.

Sioutis C, Ichalkaranje N and Jain L (2003) A framework for interfacing BDI agents to a real-time simulated environment. In *Design and Application of Hybrid Intelligent Systems: Frontiers in Artificial Intelligence and Applications* (eds Abraham A, Koppen M and Franke K). IOS Press, Amsterdam, pp. 743–748.

Sprague RH and Watson HJ (1996) *Decision Support for Management*. Prentice-Hall, Englewood Clifts, NJ.

Stathopoulou I and Tsihrintzis G (2011) Appearance-based face detection with artificial neural networks. *Intelligent Decision Technologies* **5**(2), 101–111.

Turban E and Aronson J (1998) *Decision Support Systems and Intelligent Systems*. A. Simon and Schuster Company, Upper Saddle River, NJ.

Tweedale J, Sioutis C, Phillips-Wren G, Ichalkaranje N, Urlings P and Jain L (2008) Future directions: Building a decision making framework using agent teams. In *Intelligent Decision Making: An AI-Based Approach* (eds Phillips-Wren G, Ichalkaranje N and Jain L). Springer, Berlin, pp. 387–408.

Virvou M and Tsihrintzis G (2012) Guest editorial to a special issue on 'multimedia/multimodal human-computer interaction in knowledge-based environments'. *Intelligent Decision Technologies* **6**(2), 77.

Wooldridge M (2002) *An Introduction to MultiAgent Systems*. John Wiley & Sons, Ltd, Chichester.

Woolridge M and Jennings N (1995) Agent theories, architectures, and languages: A survey. In *Intelligent Agents* (eds Wooldridge M and Jennings N), vol. 890 of *Lecture Notes in Computer Science*. Springer, New York, pp. 1–39.

Part II

INTELLIGENT TECHNOLOGIES FOR DECISION SUPPORT AND PREFERENCE MODELING

3

Designing distributed multi-criteria decision support systems for complex and uncertain situations

Tina Comes[1], Niek Wijngaards[2] and Frank Schultmann[1]

[1]*Institute for Industrial Production, Karlsruhe Institute of Technology, Germany*
[2]*Thales Research & Technology Netherlands, D-CIS Lab, The Netherlands*

3.1 Introduction

In daily life, everyone engages in decision-making. Some decisions have minor consequences and are made with little thought. Other decisions can have much greater impact and justify the effort of reflecting and deliberating on the possible options before choosing one.

If we understand decision-making as a task to select one out of several available options (called *alternatives* in the following), the question we would like to answer is: which of these alternatives is the best? To answer this question, decision analysis suggests to (i) determine the consequences of each alternative, and (ii) to evaluate and rank the alternatives on the basis of these consequences (Belton and Stewart 2002; Keeney and Raiffa 1976). To make a decision for a specific problem it is important to consider the context of the problem (Brugha 2004). The context involves the complexity of the questions that need to be addressed to determine the consequences, the uncertainty present

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

in the data and models, the number of available alternatives, the time available for the decision-making and the time until the consequences will occur (French *et al*. 2009). The cognitive and social factors to be considered comprise the number of decision-makers and stakeholders, and their respective responsibilities, skills, beliefs, attitudes towards risk, and preferences (Belton and Hodgkin 1999).

Today's decision problems are more and more characterized by their complexity (Helbing 2009; Pathak *et al*. 2007; Timmermans and Vlek 1992). Typical examples are decisions in multi-component systems such as organizations and administrations, modern supply chain networks, or policy-making (Helbing and Lämmer 2008; Levy 1994). Moreover, longer term decisions typically need to be made on the basis of incomplete and error-prone information, whose availability and quality changes over time (Rowe 1994). Together, complexity and uncertainty have several implications when trying to steer, manage, manipulate or influence any (multi-component) system, which we will outline in the following.

Complex systems are hard to understand as they behave differently from what is expected (Helbing and Lämmer 2008). Traditional paradigms, often used to support strategic decision-making, are generally based upon assumptions of linear relations and use static analyses to compare alternatives (Levy 1994). Combined, these features lead to the assumption that small changes in the data lead only to controllable changes in the result. In complex systems, which are prone to nonlinear relations, small changes can, however, lead to large fluctuations (Helbing *et al*. 2006).

The fact that in most strategic problems decision-makers pursue multiple goals adds to the complexity. The extent to which these goals have been reached is typically measured by different performance functions. Ideally, all performances should be optimized simultaneously (Fonseca and Fleming 1995; Keeney and Raiffa 1976). This is often impossible and trade-offs between different goals (such as cost vs. quality or promptness vs. precision) need to be made. If multiple decision-makers are involved or large numbers of stakeholders are affected, it is particularly important to make these trade-offs transparent to build a consensus (French *et al*. 2009).

Several studies have shown that with increasing problem complexity and amount of information, the cognitive effort required to address a problem grows and often leads to the use of heuristics: simplifying decision rules or strategies that are prone to bias and distortion and incomplete evaluation of information (Arentze *et al*. 2008; Maule and Hodgkinson 2002; Olshavsky 1979). These simplifying strategies allow complex problems to be solved without overloading the decision-makers, but often at the expense of making less than optimal choices (Maule and Hodgkinson 2002; von Winterfeldt and Edwards 1986). Computational decision support systems have been developed to retain and process large amounts of information to unburden decision-makers and reduce the effects of simplifying strategies (Comes *et al*. 2012; Maule 2010). These systems, particularly the way information is organized and visualized, should be designed to respect the way how humans manage and understand information (Maule 2010).

Uncertainty is another factor known to reduce the quality of decisions (French *et al*. 2005; Hodgkinson *et al*. 1999; Kahneman and Tversky 1979). Handling uncertainty is feasible when 'small' and controllable effects are considered or when the situation behaves more or less 'as expected' (Paté-Cornell 1996). Handling 'larger' effects with high consequences and the compounded uncertainties still poses a challenge to both humans and software systems alike, particularly when related to events that are very rare

(Ascough *et al*. 2008; Ben-Haim 2004). Uncertain information can be understood best when presented in the form of concrete and causally coherent stories, as this relates to the way humans intuitively make sense of their environment (Schoemaker 1993). The first understanding of a situation in terms of causes and effects sets the stage for the perception and integration of new information (Schoemaker 1993; Wright 2005). If the intuitive initial concepts are misleading, it is particularly important to present the new information in an appealing and convincing manner, for instance as narrative stories *(scenarios)* describing how the situation develops (Lempert *et al*. 2006). Such scenarios are also likely to appeal to the users for their compliance with most humans' preference for certainty (Kahneman and Tversky 1979; Van Schie and Van Der Pligt 1995): uncertainty is specified across single scenarios instead of within one meta-scenario that shows a distribution of results.

In this chapter we investigate how to provide decision support in complex and uncertain situations. As a stepping stone towards the identification of the key challenges that typically need to be solved, we describe three examples. Subsequently, we generalize their most important features to specify the challenges. Thereafter, we provide an overview of how our approach addresses each of them; more (technical) details can be found in the referenced publications. To tackle the challenges, our approach is based on the combination of techniques from decision analysis for making trade-offs and scenarios for 'what-if thinking'. First, these two components, multi-criteria decision analysis and scenario-based reasoning, are described. Subsequently, their integration is detailed referring to our demonstration system for decision-making in emergency management. The chapter concludes with a brief discussion on the achievement of our challenges.

3.2 Example applications

Strategic decisions can be defined as important and far-reaching decisions in terms of the actions taken, the resources committed, the number of stakeholders affected, and the impact on all future operations (Eisenhardt and Zbaracki 1992; Godet 2000). Strategic decision problems occur in fields as diverse as environmental management, policy assessment, definition of business strategies, long term project planning, or risk management (French *et al*. 2005). To illustrate typical situations, we provide three prototypical examples: a day-to-day example, an example from supply chain risk management, and an example from emergency management.

- **Choosing a smartphone.** Suppose you are interested in purchasing a new smartphone. With a plethora of offers, how do you decide which smartphone is the best for you? You may easily identify a number of smartphones that appeal to you, but choosing one remains difficult. It may be interesting to first think about which features of the smartphone are the most important for you. Should it be affordable, nicely designed, have a long battery life? All these *criteria* have an impact on how 'good' a smartphone is in your current situation. To make a decision between a number of smartphones, it is furthermore helpful to think about the way you may use your smartphone in future. Would you like to use your smartphone exclusively for phone calls, for work-related tasks such as agenda and email, or also for entertainment? What kind of support do you need? Would you like to resell the smartphone eventually? These questions are, at least partly, related to future

events, which are uncertain from your current point of view. To make a rational decision, it is important to rank the smartphones according to your criteria keeping in mind what could happen in future.

- **Supply chain management** aims at organizing the flow of goods, capital, and information between enterprises to ensure that a high level of end customer satisfaction is maintained whilst the costs are kept low. Lean and just in time production, and the access to globally dispersed suppliers and customers are a source of competitive advantages (Craighead *et al*. 2007). Coupled with these benefits are increasing complexity and uncertainty (Kleindorfer and Saad 2005). Although supply chains are usually designed to be robust against small perturbations (Helbing *et al*. 2006), disruptions may be caused by larger perturbations and extreme events (e.g., earthquakes, floods), whose frequency and severity have increased considerably in recent years (Kleindorfer and Saad 2005).

  To successfully manage supply chains it is important to pursue multiple goals. Whilst cost can relatively easily be measured, the goal of 'customer satisfaction' requires some more thought: what contributes to its achievement? Apart from price and quality, product availability, short lead times and a high service level may also be important. These measurable concretions of abstract goals are called *attributes* in the remainder of this chapter.

  The performance of a supply chain with respect to each of these goals cannot be completely controlled by decision-makers: it depends on different possible situation developments such as changes in consumer behavior, technologies, market position and competitors' behavior, prices, and occurrence of disasters. Moreover, due to the (tight) integration of industrial companies within complex supply chain networks the indirect consequences of supply chain disruptions gain in importance. Figure 3.1 illustrates the occurrence of these *cascading effects*. If supplier S_{1a}'s production is disrupted, the directly affected partners are S_2 and F (indicated by the bold arcs in Figure 3.1). The consequences of the disruption, however, propagate through the network, and in some cases the indirect consequences exceed the direct ones by more than a factor 10 (Kleindorfer and Saad 2005).

  Today's supply chain networks thus cannot be managed by only considering bilateral and static relations. Rather, information across several tiers must be

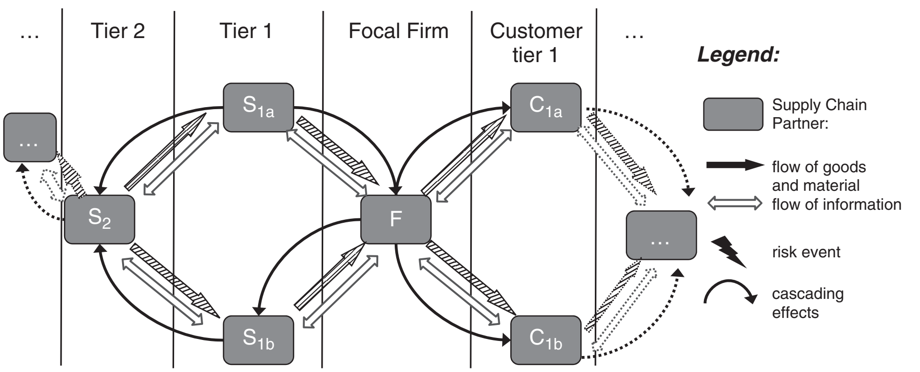

Figure 3.1 Impact of disruptions on interlaced supply networks.

continuously collected, controlled and managed. This information is, however, only available locally. Decision-making in these settings therefore requires information from various supply chain partners to be processed and assessed, coupled with a means to investigate possible failure modes, and mitigation measures by taking into account different possible future developments (Comes and Schultmann 2012).

- **Emergency management** is particularly challenging for decision-makers: most emergencies are rare and unexpected events. Typically, decisions need to be made in a relatively short time, and due to the potentially devastating consequences the pressure on decision-makers and experts is high. In these situations it is important to assess the consequences of a decision thoroughly, also when considering public post-hoc audits.

  Consider the following situation: an incident involving a gaseous hazardous chemical threatens the population in the downwind areas. Although the situation is momentarily stable by means of a temporary solution, there remains a risk that the chemical is released in large quantities, causing a plume over a densely populated area. The assessment of the consequences of preparedness measures such as evacuation of population requires that uncertainties are taken into account whilst answering some of the following questions: will the evacuation of the potentially affected areas succeed in due time? Will it cause traffic jams and accidents, or a panic? Additionally, trade-offs need to be made, for instance, is it justified to risk the health and life of fire fighters to protect the population?

  Specific expertise and skills are required to answer questions such as: 'What happens if the population is exposed to the chemical?', 'What is the concentration of the chemical over the most densely populated areas?', 'Which resources and transportation capacities are available?'. Therefore, information from various sources must be brought together to determine the consequences of each alternative. During emergency management the availability of experts is scarce, while their potential workload may be high (Comes *et al*. 2011a,b). Hence, it is particularly important to support the experts to efficiently construct scenarios that are tailored for the specific situation and to present the results in a clear manner.

In summary, the examples provided are very different, yet share some common features. Each decision-maker attempts to optimize the possible effect(s) of the decision to be made with respect to multiple goals. So, for each decision among alternatives, it is important that multiple consequences are taken into account and that, for the sake of compliance and acceptance the reasons, which led to these consequences, are presented as 'what-if stories'. Furthermore, the alternatives need to be assessed on the basis of multiple criteria in all of these stories, resulting in *robust decision-making*. Robust, in this instance, means that the consequences of an alternative are not necessarily optimal for one (best guess) story, but that the consequences in all stories are considered at least acceptable.

3.3 Key challenges

The envisioned decision support is designed for (strategic) decision-making in complex and uncertain situations. As illustrated in the examples, these decision situations show some or all of the following characteristics (Comes and Schultmann 2012; Craighead *et al*. 2007; French *et al*. 2009; Geldermann *et al*. 2009; Snowden 2005):

- *Relevance*: the decision situation can imply severe consequences that justify the effort needed for a thorough analysis of future situation developments.
- *Collaboration*: the problem can only be solved by multiple actors with heterogeneous competences, knowledge and skills and (possibly) conflicting interests.
- *Multiple objectives*: all goals, value judgments and beliefs of the decision-makers and experts and the affected stakeholders need to be considered, trade-offs need to be made.
- *Acceptance*: as the decision problem often requires the compliance of the stakeholders, transparent and understandable evaluation methods that facilitate consensus-building need to be used.
- *Imperfect information*: particularly in unexpected events (such as emergencies) or when the consequences of a decision stretch far into the future, information can be ambiguous, uncertain, conflicting, or lacking.
- *Dynamic situation development*: as time passes, the situation itself evolves, more or better information about the past or present state of the emergency may be available, and new developments need to be detected.
- *Importance of efficiency*: decisions often need to be made within a few hours or days, and experts (such as commanders in emergencies or CEOs in supply chain management) typically have to handle multiple problems at a time.
- *Decentralized setting*: experts, stakeholders and decision-makers are often geographically dispersed, and bringing all involved actors together face-to-face would require a substantial amount of time and other resources.

A decision support system needs to account for these characteristics. One of the most important aspects is the imperfect information and *uncertainty* that can be related to the data and parameters, the preferences and objectives, and, most fundamentally, to the models and assumptions (French and Niculae 2005). These uncertainties can have important consequences on the decision outcome, and it is worthwhile to consider the question: how 'wrong' can the data, preferences and models be without changing the ranking of alternatives? In situations, which are fundamentally uncertain or imply devastating consequences, it is often recommended to choose a **robust alternative** that performs sufficiently well for a broad variety of possible futures (Comes *et al*. 2011b; Schoemaker 1993; Wright 2005). Furthermore, the decision support system should help decision-makers in managing large amounts of information, analyzing and making sense of the situation and its possible developments, assessing and evaluating alternatives. Therefore, the following key challenges need to be addressed:

(1) How can information that is relevant for the problem at hand be collected and shared?

(2) How can this information be combined and processed into meaningful descriptions of the situation that are a valid basis for robust decision-making?

(3) How can the decision-making process be executed while adhering to time and effort constraints?

(4) How can decision-makers pursuing multiple goals be supported in interpreting and analyzing the results?

Our answer to these key challenges is a combination and, whenever necessary adaptation, of a set of methods and tools for making trade-offs and exploring future possible consequences. These are first described followed by an overview of our integrated approach. The discussion summarizes how each characteristic and key challenge is addressed.

3.4 Making trade-offs: Multi-criteria decision analysis

In situations with multiple objectives, multi-criteria decision analysis (MCDA) is often chosen as the basis for decision support for its coherent support in complex situations (French 1996). MCDA's popularity is mainly due to the intuitive and transparent evaluation of alternatives by refining evaluation goals in terms of a number of relatively precise but generally conflicting criteria, on which preference relations can be expressed (Stewart 1992). A view shared by many MCDA practitioners is that one of the principal benefits from the use of this well-structured approach is the learning about the problem itself and the value judgments and priorities of all involved parties contributing to an increased respective understanding (Belton and Hodgkin 1999).

3.4.1 Multi-attribute decision support

Multi-attribute decision-making (MADM) forms the basis of our approach as it allows evaluating alternatives with respect to different criteria and modeling trade-offs explicitly. Multi-attribute decision support systems include models and methods that aid the decision-makers to choose one alternative out of a (finite) list of feasible options respecting multiple criteria. That means, an alternative a_l from the set $A = \{a_1, \ldots, a_n\}$ ($n \in \mathbb{N}$) must be selected. To this end, the alternatives are ranked based on preferential information. Generally, one distinguishes compensatory and noncompensatory approaches. While compensatory approaches allow a poor performance in one criterion to be balanced by a good performance in another, this is impossible in noncompensatory approaches (Guitouni and Martel 1998).

The most common **noncompensatory approaches** are *outranking approaches*, e.g., *PROMETHEE* (Behzadian *et al*. 2010; Brans and Vincke 1985) or *ELECTRE* (Roy 1991). Outranking approaches aim at the construction of outranking relations: binary relations between pairs of alternatives that model the strength of arguments supporting the statement that an alternative a_i is at least as good as a_j *(concordance)* and the strength of the arguments against this statement *(discordance)* (Roy 1991).

Compensatory approaches can be divided into Value System and Disaggregation–Aggregation (D-A) approaches. *Value system approaches* (e.g., multi-attribute value theory, MAVT) aim at the construction of a value system that aggregates the decision-makers' preferences on the criteria based on strict assumptions on the preference relations. They require complete and transitive preference relations and the commensurability of criteria (Keeney *et al*. 1979; Siskos and Spyridakos 1999; von Winterfeldt and Edwards 1986). The elicited preferences are used as a basis to construct a unique (value or utility) function aggregating the partial preferences and performances of an alternative on multiple criteria (Siskos and Spyridakos 1999).

D-A approaches such as the utility additive (*UTA*) method aim at analyzing the decision-makers' behavior and cognitive style (Jacquet-Lagréze and Siskos 1982, 2001). In the disaggregation phase a preference model is constructed from decision-makers' judgments on a limited set of reference alternatives. These judgments are combined in the aggregation phase to value functions (Siskos and Spyridakos 1999).

Multi-attribute value theory. Due to its success in strategic decision-making (Bertsch *et al*. 2006; Chang and Yeh 2001), we use MAVT for the evaluation of alternatives. In MAVT, the decision process starts by *structuring the problem* taking it from an initial intuitive understanding to a description that facilitates quantitative analysis (von Winterfeldt and Edwards 1986). The problem structuring phase results in an *attribute tree* clustering and hierarchically ordering the decision-makers' aims, cf. Figure 3.2. The tree shows how the overall objective is divided first into less and less abstract criteria, until finally the level of attributes is reached. The attributes are a means to measure (or quantitatively estimate) the consequences arising from the implementation of any alternative a_i (Stewart 1992). The question how the consequences are modeled (i.e., how to determine the value of each attribute under the assumption that a_i is implemented) is left to the decision-makers.

The attribute tree forms the basis for the *evaluation* of each alternative. To ensure that the evaluation results in an appropriate ranking of alternatives, the decision-makers' preferences are elicited. The attribute scores are normalized to ensure the comparability of attributes, which are measured on different scales. To this end, *value functions* mapping each attribute's score to a number in [0, 1] are defined. The value functions express for each attribute how important it is to attain a performance close to the optimum (Keeney *et al*. 1979). The attribute tree is further annotated with weights that allow trade-offs to be made explicit. Each weight for a criterion $crit_j$ describes the relative importance of $crit_j$ compared with all other criteria (Raiffa 2006; von Winterfeldt and Edwards 1986). Finally, the values are aggregated to an overall score for each alternative.

MAVT results should not be understood as an imperative prescription, but as support and guidance for the decision-makers (Belton and Stewart 2002). Most of the time, the perceptions of the decision-makers will change during the decision support process

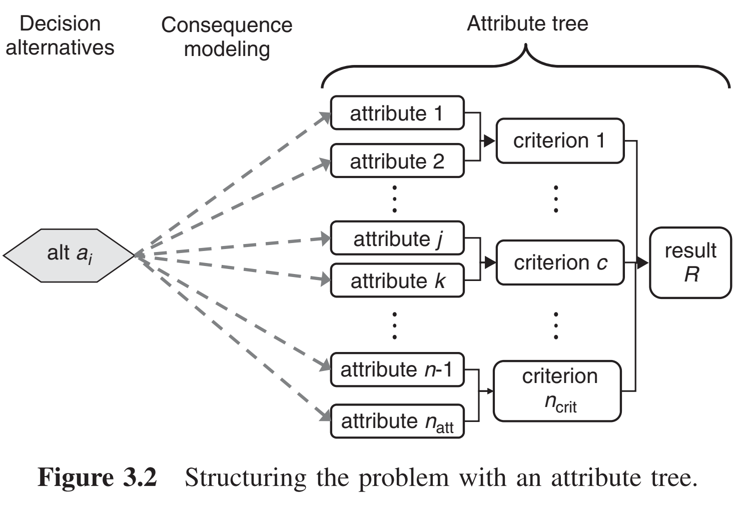

Figure 3.2 Structuring the problem with an attribute tree.

(French *et al*. 2009). Therefore, it is recommended that the modeling process is of a dynamic, cyclic nature, until a requisite decision model, whose form and content are sufficient to solve the problem, is reached (Phillips 1984).

3.4.2 Making trade-offs under uncertainty

One of the most important strengths of MAVT is the resolution of the complexity in the decision-makers' value judgments and goals by explicitly structuring the problem. As for uncertainties in parameters, data, or consequence models, MAVT assumes that all attribute scores are well-defined and known with certainty (Fenton and Neil 2001).

An approach allowing for the integration of uncertainty in the data and preferences in MAVT is performing *sensitivity analyses* (Bertsch *et al*. 2007; Ríos-Insua and French 1991). Sensitivity analyses are usually applied ex post by varying the input parameters used for the generation of an initial result (e.g., via simulation) (Saltelli *et al*. 2008). That means, sensitivity analyses are targeted at testing the stability of results to perturbations of the data and model parameters rather than exploring fundamentally different developments. Moreover, the validity of the models used is not questioned. Therefore, the applicability of sensitivity analyses in situations of fundamental uncertainty is limited.

A probabilistic approach that models uncertainties in the consequences of a decision is *multi-attribute utility theory* (MAUT). In MAUT, the consequences x_k of a decision are considered as a function of uncertain random factors with known density function $P\left(x_k\right)$. The utility function u serves as a characterization of the decision-makers' inclination towards risk (Pratt 1964). Keeney (1971) provided a theoretical framework and a set of assumptions on the preferences that allow for utility functions to be decomposed. This decomposition facilitates the preference elicitation and the aggregation of results (Fishburn 1968). Particularly, preferential and utility independence are required. *Preferential independence* implies that the preference order (indicating what is perceived as better or worse) of an attribute does *not* depend on the performances of any other attribute. *Utility independence* implies that the decision-makers' risk attitude for a given subset of attributes does not depend on the performances (and the probabilities of these performances) of the other attributes.

MAUT has been criticized for being based on unrealistic assumptions. The beliefs and preferences of the decision-makers have been considered too complex to be represented by the (quantitative) concepts of utility theory (French 1986; Gass 2005). MAUT requires the decision-makers to give judgments about preferences among imaginary bets. Therefore, Edwards (1977) argued that untutored decision-makers may either reject the whole process or accept answers suggested by the sequence of questions rather than by their own preferences.

Another problem of using MAUT is the fact that the deduction of adequate probability distributions can be problematic (Belton and Stewart 2002; Chaib-draa 2002; Jakeman *et al*. 2010). In the context of strategic decision-making it can be argued that a (purely) probabilistic conceptualization of uncertainty may not do justice to the problems involved (Hansson 1996). First, there are situations in which not all (relevant) possible outcomes of an alternative are known. Secondly, frequentist probabilities trace uncertainty back to frequencies and *repetitions*. In rare events or situations characterized by their uniqueness, it is doubtful whether the recourse to the frequency of phenomena in similar situations offers much insight into the uncertainties involved (Lempert *et al*. 2006; March and

Shapira 1987; Sigel *et al*. 2010; Wright and Goodwin 2009). Bayesian approaches using subjective probability assessments are prone to similar problems.

Due to its transparency and ease of application we adopt MAVT as the approach for the evaluation of alternatives. The inclusion of uncertainties from standard MAVT extensions as described above does not conform to our challenge to address multiple fundamentally different situation developments: scenarios. Before showing how MAVT and scenarios can be combined, the next section gives a precise definition of the term scenario and describes scenario-based reasoning processes.

3.5 Exploring the future: Scenario-based reasoning

Scenario-based reasoning (SBR) is a widely employed methodology for supporting decision-makers in complex situations (Schoemaker and Heijden 1992). Originally, scenarios were developed in response to the difficulty of creating accurate forecasts (Kahn and Wiener 1967). Generally, a scenario describes a possible situation and its development. Yet, the term has been used in a variety of ways. In the simplest case, a scenario just refers to the expected continuation of the current situation. Statements often encountered are 'under the current scenario, we anticipate that ...' (Schöpp *et al*. 1998; Van Dingenen *et al*. 2009). Another common use of the term scenario refers to a set of values for exogenous factors that enter a (simulation) model and provide the background for the actual calculations (Boonekamp 1997; Girod *et al*. 2009; Reis 2005). In our understanding one scenario is a well-structured account of *one* possible future. In essence, scenarios are dynamic stories that capture key ingredients of our uncertainty about the future of a system.

SBR employs the use of imaginary yet plausible future scenarios to help decision-makers think about the main uncertainties they face and the drivers for change; scenarios reveal the implications of current trajectories, and facilitate devising strategies to cope with those uncertainties (Montibeller and Belton 2006). Multiple scenarios offer the possibility to take into account several situation developments, which may be considered regardless of their likelihood. Therefore, SBR is particularly suitable for reasoning under severe uncertainty (i.e., in situations, when the likelihood of a series of events cannot be quantified).

Any SBR activity is purposeful: scenarios are introduced in environments of uncertainty where there is a need for action, prioritizing agendas or making decisions (Chermack 2004). Scenarios can be used in more or less formalized frameworks to raise situation awareness, to stimulate discussion and imagination, and to help broadening the horizon of the decision-makers. Basically, two distinct purposes for a scenario can be identified:

(1) Scenarios can be tailored for a decision problem. In this case the aim is assessing the consequences of a set of alternatives. The scenarios cover not only factors that can be influenced by the decision-makers, but a broad spectrum of situation developments that may be beyond the control of the decision-makers.

(2) Scenarios can be used for exploring the framework or situation in general, without being targeted at the evaluation of some predefined alternatives (raising situation awareness). This is, among other things, useful to develop promising alternatives that can later on be assessed in detail by using MCDA techniques.

3.6 Making robust decisions: Combining MCDA and SBR

This section describes the possibilities of an *integrated scenario-based MCDA approach*. Our aim is to support decision-makers in balancing the risks and chances that an alternative offers according to their risk perception and preferences. By considering how alternatives are evaluated within several scenarios the *robustness* of a decision gains in importance. This section first outlines how we understand the context of robustness before explaining the principles of scenario-based MCDA. The challenges specified in Section 3.3 are the main drivers for our new approach, see Section 3.7 for an overview of their achievement.

3.6.1 Decisions under uncertainty: The concept of robustness

Robustness is an important criterion for ‘good’ decisions under uncertainty (Ben-Haim 2004). The key principle is that decision-makers do not try to find the optimal alternative for one (best guess) scenario. By acknowledging that uncertainties abound, they aim at selecting an alternative that performs (sufficiently) well for a variety of scenarios (Matos 2007; Vincke 1999).

Figure 3.3 compares robust decisions to standard optimization techniques. For the latter, in the simplest case, the result of a decision can be determined with certainty (deterministic problem). The optimization identifies the alternative with the highest score, cf. Figure 3.3(a). The difficulty of deterministic problems consists mainly in the infinite (or very large) number of alternatives and results, which need to be compared and ranked (Rosenhead *et al*. 1972).

An extension of this approach is provided by decision analyses under (probabilistic) risk. These approaches mostly assume that a best-estimate description of the problem exists. This usually consists of a model that generates results for each alternative. Uncertainties are modeled as sets of probability distributions over the model’s input parameters (Lempert *et al*. 2006). Then, the alternatives can be ranked according to their expected utility (i.e., overall performance or ‘score’). This approach requires that all relevant uncertainties can be modeled by probability distributions of input data. Epistemic uncertainties (related to lacking knowledge and ambiguity of information) and uncertainties related to the model itself are neglected.

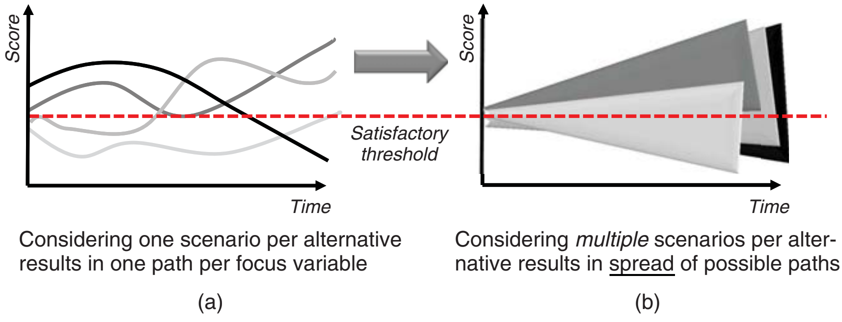

Figure 3.3 Making robust decisions by using scenarios.

Robustness has been described as a counterpart to optimization (Rosenhead *et al*. 1972). It can be understood in two ways referring to the stability or quality of results. When robustness is used to refer to the *stability* of results, it is a means to address the question of how flawed or defective the models and data can be without jeopardizing the analyses' quality (Ben-Haim 2000; Regan *et al*. 2005; Roy 2010). In the second sense, the focus is on the *quality* of an alternative: it is required that a robust alternative reaches a minimum required performance under all eventualities (Matos 2007; Vincke 1999). We follow an approach that combines both aspects: a robust decision aims at identifying an alternative that performs relatively well when compared with further alternatives across a wide range of scenarios. Figure 3.3(b) depicts this where for each alternative the performance spread in multiple scenarios is shown.

Interest in robustness has been spurred by the increasing recognition that many decisions are in fact characterized by uncertainties that cannot be quantified: severe uncertainties are present (Ben-Haim 2000; Comes *et al*. 2010; Lempert *et al*. 2006; Sluijs *et al*. 2005; van der Pas *et al*. 2010). Additionally, decision-makers, in situations that they perceive as very uncertain, often strive to identify robust alternatives (Ben-Haim 2000; Lempert *et al*. 2006; Rosenhead *et al*. 1972). In this manner, the concept of robustness is expected to aid decision-makers in better understanding the situation and its (potential) consequences.

3.6.2 Combining scenarios and MCDA

Assessing the robustness of alternatives requires development of scenarios that support decision-makers exploring the potential consequences of each alternative. This implies that mechanisms must be implemented that ensure that all relevant consequences are captured in the scenarios. Additionally, the consequences are multi-faceted, and it is usually necessary to consider multiple goals. To facilitate comparisons across both scenarios and alternatives, we use techniques from MAVT.

The decision-makers' information needs form the hub for combining scenarios and MCDA. These information needs refer to the goals of the decision-makers, as these are assumed to be the basis for the decision-making process. For the example on choosing a smartphone, these information needs could, e.g., refer to the price, battery life and design.

By using MAVT (cf. Section 3.4.1), the identification of information needs can be structured via the development of an *attribute tree* (cf. Figure 3.2). On the tree's lowest level, attributes are established that make each goal measurable.

The starting point of the decision support process is a request from the decision-makers to evaluate a set of alternatives (such as some specific types of smartphones, supply chain network configurations or emergency management plans). The scenario construction process consists of two phases, that are described in detail in the following. In the first phase, in a top-down manner *directed acyclic graphs* (DAGs) are configured that define workflows for organizing and structuring information processing and sharing. In the second phase, scenarios are generated bottom-up through the workflow by asking each expert to contribute information according to his/her skills and based on the information presented.

By merging the attribute tree and the DAG into a *decision map* (cf. Figure 3.4), scenario construction to determine a decision's (potential) consequences and the evaluation of consequences are combined. In this manner, both SBR and MCDA can benefit.

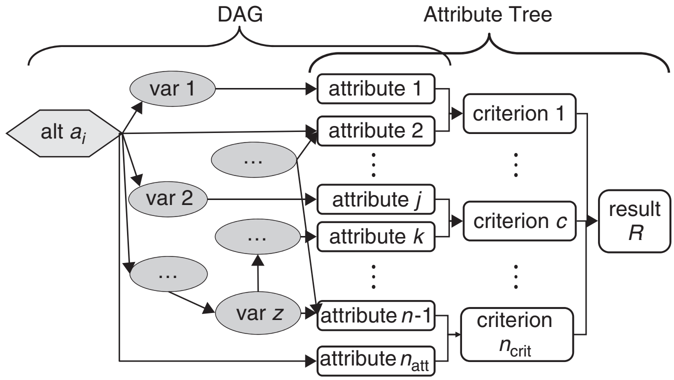

Figure 3.4 Structuring the problem with a decision map for the determination and evaluation of consequences.

For MCDA, the determination of consequences for each alternative, which are not part of the MAVT attribute tree (cf. Figure 3.2), can be made explicit. Fundamental uncertainties can be addressed by founding the decision on a broad set of scenarios. From an SBR perspective, MCDA is a means to focus and steer the scenario construction process making it more efficient and ensuring that the decision-makers' information needs are met.

3.6.3 Collecting, sharing and processing information: A distributed approach

Here, we describe our distributed approach that enables the complex overall problem of determining the consequences of a decision to be split up into a number of manageable specific sub-problems. For each of these sub-problems experts with particular knowledge and skills are identified and asked for their assessment. In that sense, our approach to scenario construction aims at establishing a cooperation between the best available experts (humans or automated systems) to construct a set of scenarios giving a broad picture of possible future developments.

As scenarios are narratives, it is important to keep track of the relations between causes and effects. DAGs are particularly suitable to represent these cause–effect chains (Galles and Pearl 1997). Therefore, we use local DAGs to represent the experts' knowledge about the interdependence of variables or the mapping between input and output information as (local) *causal maps* (CMs) (Comes *et al*. 2011a; Galles and Pearl 1997; Goodier *et al*. 2010). While sometimes causal models are referred to as mental models of the experts (Butler *et al*. 2006) or to calculate the impact of an alternative on a set of attributes (Montibeller *et al*. 2008), our approach uses CMs as a means to structure the *flow of information*, proceeding from causes to effects. Particularly, *no* standardized causal inference mechanisms are imposed, nor a specific check on grounding the causal model with reality: expert opinions on their input–output mapping are considered as de facto causal models, which are represented by DAGs.

By matching the outputs of experts to inputs of (further) experts, it becomes possible to construct an overall DAG that describes the problem (Comes *et al*. 2011b). Furthermore,

these local causal maps can be considered to be task, or service, descriptions, enabling the use of software for automatic workflow configuration.

The basis of our approach to scenario construction is a distributed approach, in which heterogeneous arbitrary processing protocols, including human-based reasoning and automated reasoning algorithms, can be combined into meaningful processing workflows. The Dynamic Process Integration Framework (DPIF) agent-based framework supports both the elicitation of the descriptions of experts and the configuration of workflows (Pavlin *et al*. 2009). This framework lets experts define their (reasoning) capabilities in terms of a task they can perform (service) and in terms of information this task requires and provides. In a task, such inputs and output are formalized, yet the algorithms involved can be arbitrary. The mapping between input and output information is assumed to be based on implicit or explicit local causal models of different types. These local models represent the experts' reasoning processes and are restricted to the form: the expert tasks each relate zero or more input information onto one output information. Moreover, although it is necessary that experts specify the type of information they need and can provide, the DPIF system allows for a flexible adaptation of the formats used (e.g., images, spoken text, number, etc.).

Once local reasoning capabilities have been defined, the DPIF framework can connect experts into meaningful processing workflows. The connection is based on the resolution of task dependencies: output of one task is used as input for one or more other tasks. In this way, collaborative reasoning can take place in the resulting workflows between specialized reasoning processes, each process focusing on a certain subset of the domain and with a respective local causal map.

The interfaces between the experts and the service-based discovery architecture of the DPIF are provided by software agents, representing the human and artificial experts. Agents also orchestrate the workflows: each agent has a set of task definitions, which are used for (i) discovery of complementary tasks required for solving problems in particular situations, (ii) delivery of specific outputs to the processes that can use them, (iii) self configuration of meaningful processing workflows and (iv) monitoring of the information flow between the tasks.

The context of the decision determines which variables are relevant, and thereby the configured workflow. For our purposes, the determination of relevant variables and their interdependencies is driven by the decision-makers' information needs that are expressed in MAVT by the attributes (cf. Section 3.4.1). Given an alternative that needs to be evaluated, the workflow configuration starts with a request to determine the attributes' values. For each request, a workflow is configured in a top-down manner. In the first step, DPIF software agents look for experts that can provide information about the attributes. When such experts are found, they refer to their local CM and indicate that in order to supply information about the attribute they depend on further input information. Again, software agents will look for further experts that can provide the needed information and so on. This process finishes when the map covers all expertise needed to evaluate all attributes. In this way, the system iteratively configures and expands a distributed reasoning model, which corresponds to the global CM.

Figure 3.5 depicts part of the workflow for the emergency management example. To determine the number of residents possibly exposed to the (potentially) released chemical, the local incident commander handling the incident on-site determines possible values of the variables vessel size, chemical and leak size. A meteorological service provides

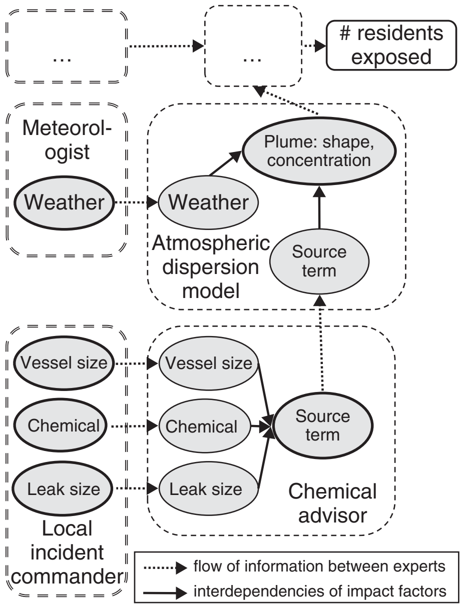

Figure 3.5 Workflow of experts for the emergency management example, showing which expert is responsible for which variables in the workflow.

weather prognosis for the next 48 h. A chemical advisor uses this information to develop a source term, which is passed on to an atmospheric dispersion model calculating the plume and concentration of the chemical in the downwind areas. By using additional information (e.g., number of residents registered in the area), finally, the number of residents possibly exposed is determined.

3.6.4 Keeping track of future developments: Constructing comparable scenarios

While the idea of using (local) expert knowledge to develop scenarios is appealing, as it allows knowledge from various sources to be integrated, the need for steering and managing the information in the scenarios arises. Particularly, it is important that the scenarios are comparable and internally fully consistent. In the distributed architecture information is processed by software agents configured in workflows without any standard DPIF-management component interfering. For the construction of consistent scenarios, this principle is adapted to a certain extent. Beyond managing the construction of scenarios, a decision-specific centralized notion is also needed to avoid information overload for experts in the workflow.

The distributed construction of scenarios is based on the workflow mechanisms described in Section 3.6.3: each agent can fulfill multiple 'jobs', where each job is reified as a DPIF-Task (Pavlin *et al*. 2010). An agent basically starts processing a Task when all the input variables values' are present (as required by its capability description).

After the top-down configuration phase, the information is processed in a bottom-up manner following the links in the global causal map. The processing of information starts

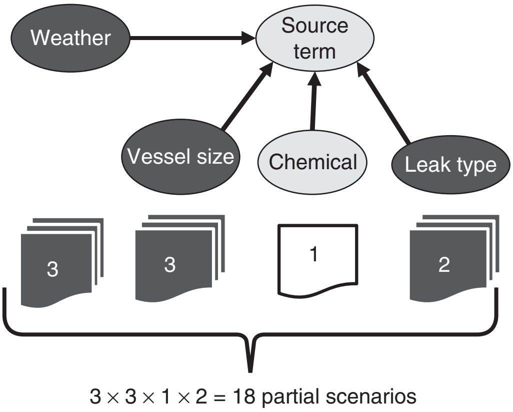

Figure 3.6 Transformation of single pieces of information to scenarios.

with the analysis of independent variables in the DAG: the variables that do not depend on input information in the workflow. In Figure 3.6 and Figure 3.7, for instance, the independent variables are *Source term, Weather, Vessel size,* and *Chemical*. The experts providing information about these variables assess the states of the corresponding variables based on their local knowledge. If there is uncertainty about the state of a variable, an expert may pass on several possible estimates for the variable. This information can be encoded as a set of numerical values, a number of maps, some text files and so on, but all output variants must fit the input that was used to determine these estimates. The states of subsequent nodes that depend on single- or multiple-valued input nodes must be determined taking into account all the states of the input nodes. These evaluation iterations are performed following all the links in the causal map until the attributes are

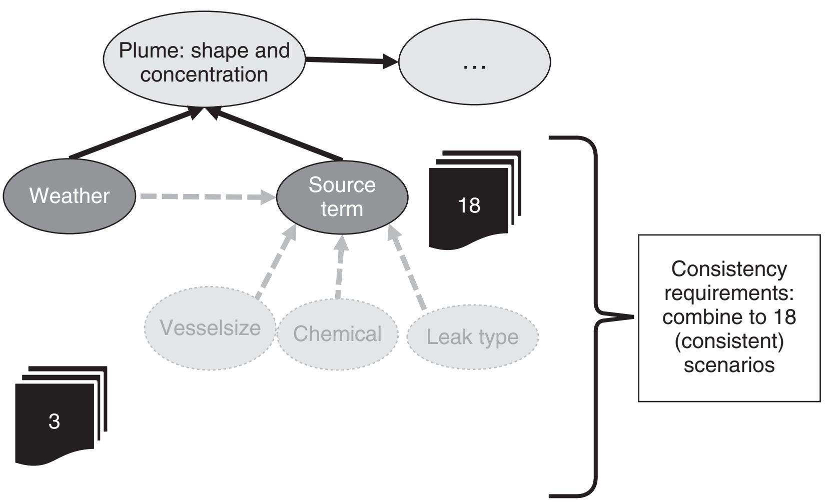

Figure 3.7 Procedure for ensuring scenario consistency.

reached. At each node, the expert, who agreed to provide results given the information received, uses his/her local reasoning process to determine the possible state(s) of the corresponding variable. In this way, a number of scenarios is created, which each correspond to a given instantiation of all variables in the global CM. The arising scenarios can be understood as a way of expressing uncertainty reflected in a range of possible and relevant states for each variable.

Ensuring consistency. An important aspect of scenario construction is to safeguard the consistency in each scenario and even for all partial scenarios. Partial, or incomplete, scenarios are scenarios that do not provide values for all attributes. The structure of the DAG is exploited to ensure consistency.

Figure 3.6 shows a part of the scenario construction workflow for the emergency management example. Here, the *Source term* is calculated on the basis of information about the *Weather*, the *Vessel size*, the *Chemical*, and the *Leak type*. As there is uncertainty about the variables in dark grey, the respective experts specify three possible values for *Weather* and *Vessel size*, and two for the *Leak type*, while the *Chemical* is known with certainty (one value). Before referring to the expert calculating the source term (volume of chemical released over time), a scenario management agent must combine these values to *partial* (incomplete) and consistent scenarios. Figure 3.6 shows that the variables are independent (i.e., no path leads from one variable to any of the others). Hence, all values for *Weather* need to be combined with all values of *Vessel size* and the *Leak type*. In this manner, $3 \times 3 \times 1 \times 2 = 18$ incomplete (or partial) scenarios arise. On the basis of these scenarios, 18 possible source terms (one per scenario) are determined.

In Figure 3.7, the scenario construction process continues with the *Plume* calculation. The atmospheric dispersion model, which performs the computing, needs input information about the *Source term,* and the *Weather* (e.g., on precipitation and air pressure). Here, 3 possible weather situations need to be combined with 18 possible source terms. *Weather* and *Source term* are causally related, and simple combinatorics are *not* allowed: this generates inconsistent partial scenarios (containing values that experts have never combined into one scenario). Detecting this entanglement is not tractable from the perspective of an individual expert: the causal structures in the first step (depicted as dotted lines in Figure 3.7) are not visible for the experts later in the network. Hence, mechanisms that keep track of the scenario construction process and ensure consistency are implemented within the software agents.

When an agent produces one output for a specific Task, then those values are passed on to other agents who need this output information. When an agent produces multiple outputs for a specific task, then a special ‘cloning’ operation starts: further in the workflow (following the flow of information) all dependent tasks are cloned (‘copied’). The cloned tasks are created to ensure that each task has always the same output, since the output of a task is not allowed to change during scenario-based reasoning. Through cloning tasks new tasks can be created to handle other outputs as well.

On the basis of this cloning operation, a mechanism has been designed to guarantee consistent scenarios. A metaphor for this mechanism is perhaps best envisioned as creating ‘virtual private workflows’, one private workflow per scenario under construction. ‘Virtual’, as there is only one ‘real’ workflow configured with the agents, and ‘private’ as values from one virtual private workflow are not allowed to ‘cross over’ with values in another virtual private workflow. This metaphor goes slightly awry when discussing *scenario merging*, which depends on the workflow structure (Comes 2011).

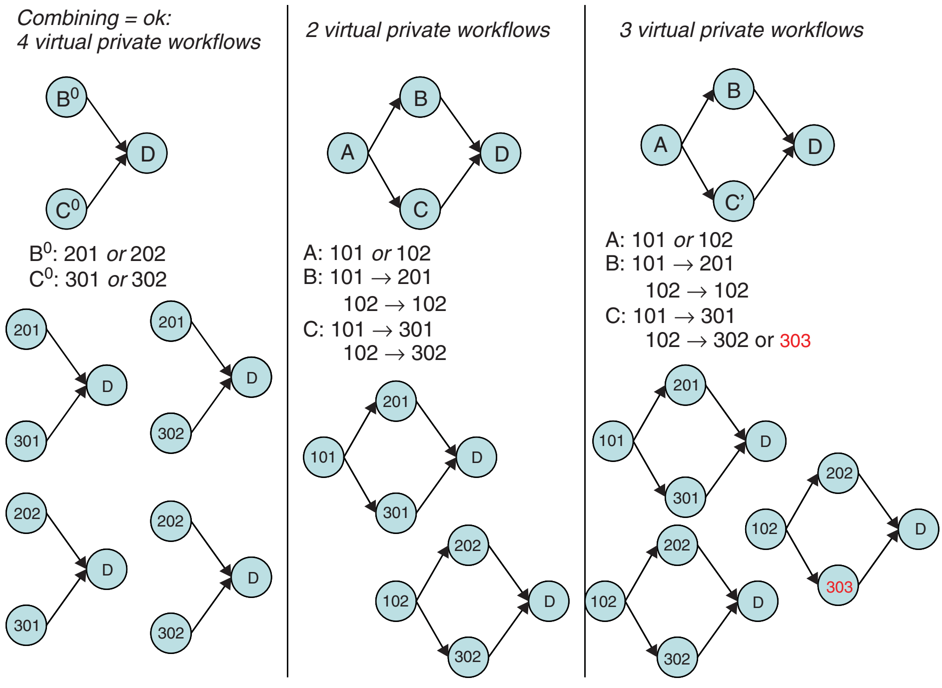

Figure 3.8 Example of the process used to construct consistent scenarios.

Figure 3.8 provides a brief example, showing per scenario a consistent (in terms of values of variables) virtual private workflow.

3.6.5 Respecting constraints and requirements: Scenario management

Ideally, the set of scenarios covers all possible situation developments (Haimes *et al*. 2002; Kaplan 1997; Kaplan and Garrick 1981). In large and complex problems it is, however, often impossible to identify all scenarios as complex problems preclude the description of interdependencies between the variables by a unique model (e.g., a system of equations). Particularly, no simple filtering approach (e.g., specification of boundary conditions) for identifying the admissible scenarios can be applied.

By using the scenario construction principles explained earlier, consistent scenarios arise. Growing complexity and uncertainty are, however, reflected in a growing number of variables and a growing number of values per variables. Therefore, the more complex and uncertain a situation is, the more (partial) scenarios arise. A requirement for the implementation and application of the SBR approach is curbing the potential ‘combinatorial’ explosion of the number of scenarios. To avoid information overload and too high workload, approaches to filter the most *relevant* scenarios need to be implemented. These approaches comprise mechanisms to prune falsified scenarios, select the most relevant scenarios and update scenarios, about which new information is available. As the scenario selection is the basis for scenario management, we explain its principles. Details about scenario pruning and updating can be found in (Comes 2011; Comes and Schultmann 2012).

Measuring scenario similarity. When comparing two scenarios, one of the most fundamental questions is: are they equal? This is of particular interest for scenario management, for if they are equal, it is sufficient to consider only one of both scenarios. A natural and useful extension of equality is *similarity*: how 'close to equal' are the scenarios? The idea is to group together scenarios that are similar with respect to the consequences they represent. To measure the difference in terms of the consequences of relevance, we refer once more to techniques of MADM that enable exploring the space of possible *evaluations*.

Figure 3.9 shows the situation for incomplete scenarios: the scenarios do not yet contain sufficient information, and further experts need to contribute. Ideally, the scenarios cover the space of all possible consequences. For illustrative purposes, the large gray circle represents a visualization of the consequences in terms of two dimensions (criteria). As only a finite number of scenarios can be constructed the surface of the circle can, however, not be fully explored through scenarios. Additionally, the experts working on the current set of partial scenarios (further in the workflow) have only a limited processing capacity and can only contribute to n scenarios. The aim is therefore to identify the n most distinct scenarios with respect to the evaluation. To this end, the scenarios are clustered into groups. As the scenarios are incomplete, the evaluation of alternatives cannot be assessed directly through values of attributes. In these situations, the similarity of scenarios can also be based on variables in the scenarios that are not the attributes. That is, some of the variables are 'indicative' of a possible consequence, and hence can be used to assess similarity of partial scenarios (those scenarios have a value for these indicator variables). Based on indicator similarity, again groups of scenarios can be established and a representative chosen for continuation.

After having established the groups of similar scenarios (cf. Figure 3.9, where the scenarios are grouped into 6 clusters), per group one scenario is chosen and further processed or analyzed. This *representative scenario* (shown in dark gray in Figure 3.9) is meant to prototypically represent the situation development's consequences. The use of single scenarios representing each class of similar scenarios (vs. using sets of scenarios or an aggregation of the results only) is supported by investigations on the perception of risk. These investigations confirmed a considerable difference in the attitude towards

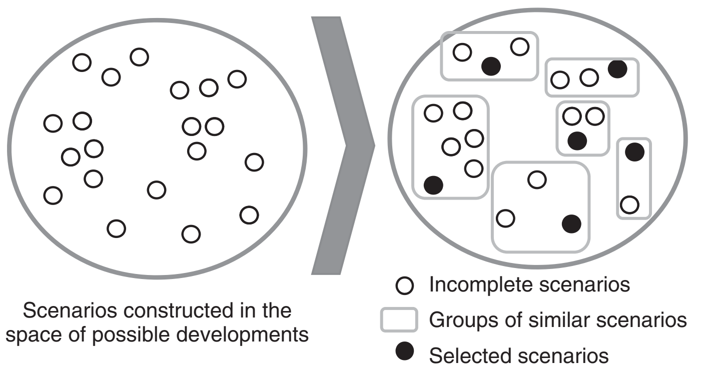

Figure 3.9 Scenario selection principles.

risk when actual scenarios were presented in terms of an understandable what-if story instead of a more aggregated and less intuitively appealing description (Benartzi and Thaler 1999).

Respecting time and effort constraints. By exploiting the structure of the DAG, inspecting the currently constructed (partial) scenarios, and establishing their similarity, it becomes possible to also predict, and manage, the desirable number of (partial) scenarios to be further processed in the remainder of the workflow. When experts are capable of providing effort and time estimates for conducting their Tasks, it becomes feasible to further discriminate among partial scenarios to select the number of partial scenarios that are most dissimilar and can still be continued by the remainder of the workflow. By conducting these assessments repeatedly during scenario construction, we 'continuously' manage the scenario construction process.

Scenario selection. Using these measures for prioritisaton, (partial) scenarios can be identified to be constructed 'first.' This implies that the decentralized scenario construction mechanism must be coupled with a scenario management mechanism which influences the agents' autonomy in working on their Tasks. A control service has been implemented that monitors the scenario construction in the workflow of agents for our emergency management example. This control service 'listens' to all the Task cloning messages, discovers agent workflow dependencies, and extracts scenarios from the task communication messages. Scenario information is placed in the appropriate data structures, and subsequently our Matlab SBR and MCDA evaluation is executed to obtain evaluation results and scenario prioritizations that are communicated to agents to prioritize their open Tasks.

In essence, we are introducing a 'decision-centric' view on the software agents (and thus also the human experts) participating in the workflow for assisting a decision-maker: these agents thereby agree to additional protocols and mechanisms to ensure correct processing of information and reducing a potentially very large number of scenarios to a workable subset. This perspective is an open question for future research, including the careful analysis of possible adverse impacts on decentralized information processing and agent autonomy principles.

3.6.6 Assisting evaluation: Assessing large numbers of scenarios

After the scenario generation has been completed, the alternatives are evaluated using the attribute tree. In the simplest case, all variables are deterministic, and only one (complete) scenario $S\left(a_l\right)$ with a corresponding set of attribute scores for each alternative $a_l \in A$ is derived. Hence, standard MAVT techniques as described in Section 3.4.1 can be applied. If there are uncertain variables, a set of (complete) scenarios $SS\left(a_l\right)$, $\left|SS\left(a_l\right)\right| > 1$ is created for each alternative. Analogous to the deterministic case, each scenario $S_i\left(a_l\right) \in SS\left(a_l\right)$ is evaluated using MAVT techniques.

3.6.6.1 Comparing single scenarios: Exploring the stability of consequences

Scenarios, when presented as stories, are a means to provide transparent and easily understandable decision support. Although a single scenario cannot convey the information which is present in the full set of scenarios, it can be important to present some

specific and detailed results, which are unexpected and challenge the current world view, in detail to the decision-maker.

As before, it is important to select the most relevant scenarios, i.e., the scenarios which enable a good exploration of the space of possible developments. These can, for instance, be scenarios that represent an unexpected outcome. Potentially, insights into *why* an alternative performs unexpectedly good or bad can be fostered. If there is sufficient time, these insights can be used for refining and adapting the alternatives in an iterative process.

Often, for decisions under severe (non quantifiable) uncertainty, it is recommended to consider the worst and best cases. As probabilities for the scenarios cannot be established, worst and best scenarios represent the spectrum of what can happen (Schoemaker 1993). Decision-makers are then asked to choose their *risk aversion* level, i.e., the importance they want assign to the worst case, and to base the decision thereon. By using the MAVT evaluation as a means to reduce complexity, it becomes possible to justify the choice of worst and best evaluated scenarios on the basis of the decision-makers' actual preferences.

In this manner, our approach supports decision-makers in choosing an alternative whose total performance (or performance in selected criteria) does not fall below a threshold τ, reflecting the minimal required performance. Thus, a robust decision in the following sense is supported: an alternative that performs sufficiently well for a set of scenarios (Ben-Haim 2000) or that guarantees that a minimum performance is reached for all scenarios (Vincke 1999) can be chosen (see also Figure 3.3).

3.6.6.2 Considering multiple scenarios: Aggregation techniques

To integrate the results for all scenarios and to avoid cognitive biases the results can be *aggregated* (Comes *et al*. 2012). To enable taking into account the decision-makers' preferences, we use MAVT techniques on the results of single scenarios to complement the scenario selection.

The aggregation of the scenarios' performances $R\left(S_i\left(a_l\right)\right)$ to the performance $R\left(a_l\right)$ of an alternative a_l can be achieved additively or multiplicatively, depending on the preferences of the decision-makers (independence considerations, cf. Section 3.4.2). Additive aggregation (e.g., the simple additive weighting (SAW) method) is prevalent due to its intuitive understandability that makes it easily accessible to those involved in the decision-making process (Belton and Stewart 2002). Nevertheless, the potential error of using additive aggregation must be compared with its advantages. Stewart (1996) showed that the sensitivity of results to errors that may be caused by an undue use of the additive aggregation is significantly smaller than the sensitivity to incorrect modeling of the value functions. Therefore, we apply SAW. For each alternative a_l, we consider the set of scenarios that contain the alternative $SS\left(a_l\right)$. The performance of a_l can be calculated by:

$$R\left(a_l\right) = \sum_{S_i\left(a_l\right) \in SS\left(a_l\right)} \omega\left(S_i\left(a_l\right)\right) \cdot R\left(S_i\left(a_l\right)\right)$$

As before, the decision-makers' preferences ω need to be determined. Analogous to the approach for the criteria, the question is how important each scenario is with respect to all other scenarios. This question is, however, difficult to answer for each single scenario if the number of scenarios is large. In the following, we present two

techniques to support the preference elicitation: the first technique is based on the concept of satisficing, the second makes use of the ranking of scenario evaluations.

Weighting based on satisficing. This approach relies on the goal-oriented idea of robustness (Matos 2007; Vincke 1999), which corresponds to Simon's concept of satisficing (Simon 1979). This concept is based on the assumption that decision-makers will not choose the alternative that maximizes the performance but the one that guarantees satisfactory performance with respect to all criteria. This satisfactory performance can, in MAVT, be modeled by a set of thresholds ξ_j, where for each criterion j, the performance of an alternative must be better than ξ_j.

If the scenarios represent a very broad span of possible futures, however, there can be situations when all alternatives violate some constraints. Then, an alternative that is *close* to satisfactory performance in all scenarios is sought. To develop a measure for the distance to the satisfactory performance, we allow preferences and risk aversion to vary between criteria. For example, threats to human health and safety may be treated rather conservatively, while the performance of criteria reflecting economic losses or resource use may be handled in a rather risk-neutral way.

To apply this approach, the decision-makers need to define thresholds that should be respected in any scenario. If in a scenario S_j a threshold is violated, a penalty function is used to increase the weight of S_j compared with a basic weight. The penalty function conveys the decision-makers' perception of how important a threshold violation is.

Weighting based on scenario ranking. An alternative to the weighted sum aggregation is the ordered weighted averaging (OWA) operator (Yager 2002, 2008). By using the OWA, the performance of alternative a_l is calculated by:

$$R\left(a_l\right) = \sum_{S_i\left(a_l\right) \in SS\left(a_l\right)} \omega\left(S_i\left(a_l\right)\right) \cdot \sigma\left(R\left(S_i\left(a_l\right)\right)\right)$$

The main difference between the weighted sum and the OWA approach is the permutation operator σ: before weighing the results, they are ordered such that $R_1 \leq R_2 \leq \ldots \leq R_n$. The weighted sum takes into account the importance of each scenario regardless of the results in the other scenarios. Contrarily, the application of the OWA operator is based on the importance of a value (of a criterion) in *relation* to other values. Thus, the OWA permits weighting the values in relation to their ordering (Torra 1997): decision-makers can express their preferences in relation to the ranking of the scenarios, but not the preferences between single scenarios (Valls *et al*. 2009).

To determine the OWA weights two measures are introduced (Yager 1988): the dispersion and the attitudinal character. The dispersion essentially measures the degree to which all or just some arguments influence the result of the aggregation (O'Hagan 1988; Yager 1988). For the dispersion, we used the *Shannon entropy*, that is the most commonly used dispersion measure (Malczewski 2005; Xu 2005; Yager 2008). Contrarily, the attitudinal character can be interpreted as a measure of the degree of the decision-maker's optimism (Yager 1988), which is captured by emphasizing the best or worst evaluated scenarios' results.

Which method to choose? The satisficing approach allows the decision-makers' preferences to be precisely modeled. Although this approach reduces the workload for the decision-makers compared with direct weighting methods, it is demanding. The determination of weights by the scenario ranking is less demanding, since the decision-makers

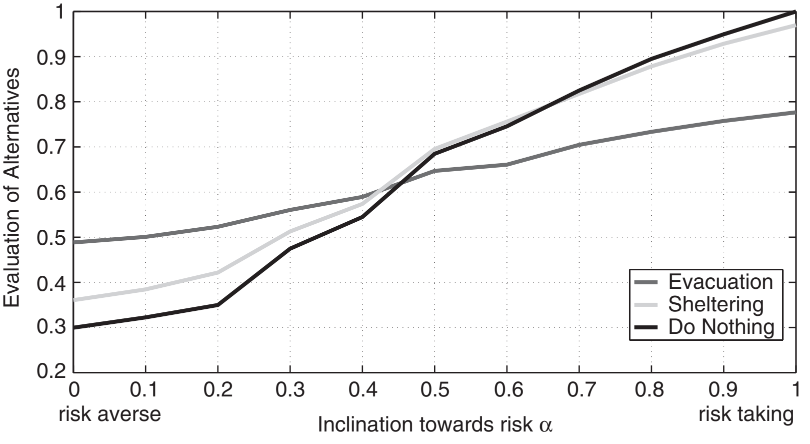

Figure 3.10 Results for emergency management example and varying levels of risk aversion.

need to specify one parameter only. As this risk averison parameter α is a rather abstract construct, we suggest to perform sensitivity analyses before making a decision. Which method to choose, depends on the time available for the elicitation of weights (in general, the method based on satisficing will require more time), the preferences and experience of the decision-makers in MCDA and related fields.

Figure 3.10 shows such a sensitivity analysis for the emergency management example by varying the inclination towards risk. The consequences of three alternatives were investigated: evacuation or sheltering of the downwind areas, or relying on the success of the transfer (no release of the chemical) and not implementing any preparedness measure (do nothing).

3.7 Discussion

This chapter started with the identification of key challenges for decision support in complex and uncertain situations. While comparisons with related work have been made throughout the chapter, this section summarizes how our approach matches with the characteristics, notions of robust decision-making, and key challenges described in Section 3.3.

Our approach accounts for the characteristics of decision support for (strategic) decision-making as follows:

- *Relevance*: our approach can be applied when the decision is sufficiently important, complex or uncertain – the justification of spending effort comes from the decision-makers, not our approach. Our approach does provide a means to support a decision-maker in a resource-bound setting (see *Efficiency*).

- *Collaboration*: our scenario construction process explicitly incorporates a 'decision'-specific workflow of experts that together assess the consequences of applying an alternative.

- *Multiple objectives*: our approach involves MAVT evaluations, which provides a structured means to make trade-offs between multiple conflicting criteria.
- *Acceptance*: our approach involves MAVT evaluations *per* scenario, the possibility to select scenarios and a multi-scenario evaluation; all these techniques are based on preferences elicited from the decision-makers.
- *Imperfect information* and *dynamic situation development*: our approach captures uncertainty with scenarios, while also providing mechanisms for scenario updating and pruning.
- *Efficiency*: our approach incorporates scenario management, in which time and effort constraints are taken into account to steer scenario construction towards sufficiently *different* scenarios to support robust decision-making.
- *Decentralized setting*: our approach uses ICT systems to bring together information from locally dispersed experts. It is based on an agent-based framework, which facilitates distributed information processing. We combined this system with components for orchestrating decision-centric coherent workflows and scenario construction.

Our approach is geared to support *robust decision-making*.

- A set of alternatives is determined, often including the option 'do nothing'.
- Multiple scenarios are are constructed *for each* alternative. Each scenario captures a possible future development of the situation.
- Alternatives are first evaluated using MAVT for each of their scenarios.
- Subsequently, the overall evaluation of an alternative a_l is based on an aggregation of the evaluations of all scenarios that assume the implementation of a_l. The evaluation is combined with an assessment of the relative qualities of the scenarios.
- The evaluations are conducted using preferences from the decision-makers taking into account their beliefs, requirements and attitudes towards risk.
- The result is a recommendation for an alternative that scores 'best', given the constructed scenarios and preferences. This result is complemented by a transparent overview of why and how this evaluation was achieved.

Given enough time for discussion and reflection, these results can be used to refine and adapt the alternatives on the basis of the insights gained about their strengths and weaknesses.

The key challenges are addressed as follows:

(1) *How can information that is relevant for the problem at hand be collected, shared and processed?* Our approach employs decision-centric workflows that are constructed to fulfill the decision-makers' information needs, which are identified using MAVT problem structuring techniques. Together with the decision-context information and the alternatives to be evaluated, these needs are the beginning of the workflow. The means-ends workflow construction only involves additional experts when their expertise is on a 'path' leading to one or more of the evaluation

'attributes.' By requiring each expert to specify the information needed to perform his/her task, processing of irrelevant and redundant information is reduced.

(2) *How can this information by combined to meaningful descriptions of the situation that enable robust sense or decision-making?* The processing of the workflow of experts is orchestrated to construct internally consistent (partial) scenarios, whenever an expert produces more than one result for a task.

(3) *How can the decision-making process be executed while adhering to time and effort constraints?* Our approach features an explicit 'decision-centric' scenario management mechanism, that guides the scenario construction process to focus on those scenarios that are sufficiently different to foster robust decision-making.

(4) *How can decision-makers pursuing multiple goals be supported in interpreting and analyzing these descriptions?* The decision-makers' goals are captured along the perspectives: multi-criteria evaluation of alternatives, qualities of scenarios and resource constraints (time, effort) on the decision-making process.

The concepts described in this chapter outline the contours of our approach; ultimately, the proof of the pudding is in the eating. Within the Diadem FP7 project a demonstration system has been implemented for robust decision-making on the basis of the emergency management use case. This demonstration system illustrates that our approach can be operationalized and is feasible.

3.8 Conclusion

This chapter describes our approach to the design of a system supporting *robust decision-making* in complex and uncertain situations. We outlined how a distributed scenario-based multi-criteria decision support system can be constructed that answers key challenges such as efficient information processing, transparent evaluations and the consideration of constraints in terms of resources and time available. Although this chapter provides an overview, detailed descriptions of our approach are available elsewhere. We have shown that we have addressed the identified key challenges for and characteristics of decision support systems for complex and uncertain situations. Furthermore, the feasibility of our approach is shown by the accompanying demonstration system. This leads to the overall conclusion that we have succeeded in designing our decision support system for robust decision-making.

Further research is required to study the *construction* of possible alternatives and other refinements of the current approach such as investigating visualization techniques and extending our approach to multi-criteria scenario-based support for planning and management, e.g., in the risk assessment and supply chain management domain.

Acknowledgment

The work presented in this chapter is partially based on research conducted within the Diadem project (2008–2011), which was funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D, ref. no. 224318, www.ist-diadem.eu.

References

Arentze TA, Dellaert BGC and Timmermans HJP (2008) Modeling and measuring individuals' mental representations of complex spatio-temporal decision problems. *Environment and Behavior* **40**(6), 843–869.

Ascough J, Maier H, Ravalico J and Strudley M (2008) Future research challenges for incorporation of uncertainty in environmental and ecological decision-making. *Ecological Modelling* **219**(3–4), 383–399.

Behzadian M, Kazemzadeh R, Albadvi A and Aghdasi M (2010) PROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research* **200**(1), 198–215.

Belton V and Hodgkin J (1999) Facilitators, decision makers, D.I.Y. users: Is intelligent multicriteria decision support for all feasible or desirable? *European Journal of Operational Research* **113**(2), 247–260.

Belton V and Stewart T (2002) *Multiple Criteria Decision Analysis. An Integrated Approach*. Kluwer Academic Publishers, Boston.

Ben-Haim Y (2000) Robust rationality and decisions under severe uncertainty. *Journal of the Franklin Institute* **337**(2–3), 171–199.

Benartzi S and Thaler RH (1999) Risk aversion or myopia? Choices in repeated gambles and retirement investments. *Management Science* **45**(3), 364–381.

Ben-Haim Y (2004) Uncertainty, probability and information-gaps. *Reliability Engineering & System Safety* **85**(1–3), 249–266.

Bertsch V, Geldermann J and Rentz O (2007) Preference sensitivity analyses for Multi-Attribute Decision Support. In *Operations Research Proceedings 2006*. Springer, Berlin, pp. 411–416.

Bertsch V, Geldermann J, Rentz O and Raskob W (2006) Multi-criteria decision support and stakeholder involvement in emergency management. *International Journal of Emergency Management* **3**(2–3), 114–130.

Boonekamp P (1997) Monitoring the energy use of households using a simulation model. *Energy Policy* **25**(7–9), 781–787.

Brans JP and Vincke P (1985) A Preference Ranking Organisation Method: The PROMETHEE Method for Multiple Criteria Decision-Making. *Management Science* **31**(6), 647–656.

Brugha CM (2004) Structure of Multi-Criteria Decision-Making. *The Journal of the Operational Research Society* **55**(11), 1156–1168.

Butler JC, Dyer JS and Jia J (2006) Using attributes to predict objectives in preference models. *Decision Analysis* **3**(2), 100–116.

Chaib-draa B (2002) Causal maps: theory, implementation, and practical applications in multiagent environments. *IEEE Transactions on Knowledge and Data Engineering* **14**(6), 1201–1217.

Chang YH and Yeh CH (2001) Evaluating airline competitiveness using multiattribute decision making. *Omega* **29**(5), 405–415.

Chermack TJ (2004) Improving decision-making with scenario planning. *Futures* **36**(3), 295–309.

Comes T (2011) *Decision Maps for Distributed Scenario-Based Multi Criteria Decision Support*. PhD thesis, Karlsruhe Institute of Technology.

Comes T, Hiete M, Wijngaards N and Schultmann F (2010) Enhancing robustness in multi-criteria decision-making: A scenario-based Approach. In *Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems*. IEEE Computer Society, Washington, DC, pp. 484–489.

Comes T, Hiete M, Wijngaards N and Schultmann F (2011a) Decision Maps: A framework for multi-criteria decision support under severe uncertainty. *Decision Support Systems* **52**(1), 108–118.

Comes T and Schultmann F (2012) Enhancing robustness against natural hazards in supply chain management. *10th International Conference on Applied Mathematical Optimization and Modelling (APMOD 2012)*.

Comes T, Wijngaards N, Hiete M, Conrado C and Schultmann F (2011b) A distributed scenario-based decision support system for robust decision-making in complex situations. *International Journal of Information Systems for Crisis Response and Management* **3**(4), 17–35.

Comes T, Wijngaards N, Maule J, Allen D and Schultmann F (2012) Scenario reliability assessment to support decision makers in situations of severe uncertainty. *IEEE Conference on Cognitive Methods in Situation Awareness and Decision Support*. IEEE Computer Society, Washington, DC, pp. 30–37.

Craighead CW, Blackhurst J, Rungtusanatham MJ and Handfield RB (2007) The severity of supply chain disruptions: Design characteristics and mitigation capabilities. *Decision Sciences* **38**(1), 131–156.

Edwards W (1977) How to use multiattribute utility measurement for social decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics* **7**(5), 326–340.

Eisenhardt KM and Zbaracki MJ (1992) Strategic decision making. *Strategic Management Journal* **13**, 17–37.

Fenton N and Neil M (2001) Making decisions: using bayesian nets and MCDA. *Knowledge-Based Systems* **14**(7), 307–325.

Fishburn PC (1968) Utility theory. *Management Science* **14**(5), 335–378.

Fonseca C and Fleming P (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* **3**(1), 1–16.

French S (1986) *Decision Theory: An Introduction of the Mathematics of Rationality*. Ellis Harwood Limited, Chichester.

French S (1996) Multi-attribute decision support in the event of a nuclear accident. *Journal of Multi-Criteria Decision Analysis* **5**(1), 39–57.

French S, Bedford T and Atherton E (2005) Supporting ALARP decision making by cost benefit analysis and multiattribute utility theory. *Journal of Risk Research* **8**(3), 207–223.

French S, Maule J and Papamichail N (2009) *Decision Behaviour, Analysis and Support*. Cambridge University Press, Cambridge.

French S and Niculae C (2005) Believe in the model: mishandle the emergency. *Journal of Homeland Security and Emergency Management* **2**(1).

Galles D and Pearl J (1997) Axioms of causal relevance, *Artificial Intelligence* **97**(1–2), 9–43.

Gass SI (2005) Model world: The great debate-MAUT versus AHP. *Interfaces* **35**(4), 308–312.

Geldermann J, Bertsch V, Treitz M, French S, Papamichail KN and Hämäläinen RP (2009) Multi-criteria decision support and evaluation of strategies for nuclear remediation management. *Omega* **37**(1), 238–251.

Girod B, Wiek A, Mieg H and Hulme M (2009) The evolution of the IPCC's emissions scenarios. *Environmental Science & Policy* **12**(2), 103–118.

Godet M (2000) The art of scenarios and strategic planning: Tools and pitfalls. *Technological Forecasting and Social Change* **65**(1), 3–22.

Goodier C, Austin S, Soetanto R and Dainty A (2010) Causal mapping and scenario building with multiple organisations. *Futures* **42**(3), 219–229.

Guitouni A and Martel JM (1998) Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research* **109**(2), 501–521.

Haimes YY, Kaplan S and Lambert JH (2002) Risk filtering, ranking, and management framework using hierarchical holographic modeling. *Risk Analysis* **22**(2), 383–398.

Hansson SO (1996) Decision making under great uncertainty. *Philosophy of the Social Sciences* **26**(3), 369–386.

Helbing D (2009) Managing complexity in socio-economic systems. *European Review* **17**, 423–438.

Helbing D, Ammoser H and Kühnert C (2006) Disasters as extreme events and the importance of network interactions for disaster response management. In *Extreme Events in Nature and Society* (eds Albeverio S, Jentsch V and Kantz H), The Frontiers Collection. Springer, Berlin, pp. 319–348.

Helbing D and Lämmer S (2008) Managing complexity: An introduction. In *Managing Complexity: Insights, Concepts, Applications* (ed. Helbing D), vol. 32 of *Understanding Complex Systems*. Springer, Berlin, pp. 1–16.

Hodgkinson GP, Bown NJ, Maule J, Glaister KW and Pearman AD (1999) Breaking the frame: an analysis of strategic cognition and decision making under uncertainty. *Strategic Management Journal* **20**(10), 977–985.

Jacquet-Lagréze E and Siskos Y (1982) Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research* **10**(2), 151–164.

Jacquet-Lagréze E and Siskos Y (2001) Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research* **130**(2), 233–245.

Jakeman J, Eldred M and Xiu D (2010) Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics* **229**(12), 4648–4663.

Kahn H and Wiener AJ (1967) The next thirty-three years: A framework for speculation. *Daedalus* **96**(3), 705–732.

Kahneman D and Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica* **47**(2), 263–291.

Kaplan S (1997) The words of risk analysis. *Risk Analysis* **17**(4), 407–417.

Kaplan S and Garrick BJ (1981) On the quantitative definition of risk. *Risk Analysis* **1**(1), 11–27.

Keeney RL (1971) Utility independence and preferences for multiattributed consequences. *Operations Research* **19**(4), 875–893.

Keeney RL and Raiffa H (1976) *Decisions with Multiple Objectives*. John Wiley & Sons, Ltd, New York.

Keeney RL, Raiffa H and Rajala DW (1979) Decisions with multiple objectives: Preferences and value trade-offs. *IEEE Transactions on Systems, Man and Cybernetics* **9**(7), 403.

Kleindorfer PR and Saad GH (2005) Managing disruption risks in supply chains. *Production and Operations Management* **14**(1), 53–68.

Lempert RJ, Groves DG, Popper SW and Bankes SC (2006) A general, analytic method for generating robust strategies and narrative scenarios. *Management Science* **52**(4), 514–528.

Levy D (1994) Chaos theory and strategy: Theory, application, and managerial implications. *Strategic Management Journal* **15**(S2), 167–178.

Malczewski J (2005) Integrating multicriteria analysis and geographic information systems: the ordered weighted averaging OWA approach. *International Journal of Environmental Technology and Management* **6**(13), 7–19.

March JG and Shapira Z (1987) Managerial perspectives on risk and risk taking. *Management Science* **33**(11), 1404–1418.

Matos MA (2007) Decision under risk as a multicriteria problem. *European Journal of Operational Research* **181**(3), 1516–1529.

Maule A (2010) Can computers overcome limitations in human decision making? *International Journal of Human Computer Interaction* **26**, 108–119.

Maule A and Hodgkinson G (2002) Heuristics, biases and strategic decision making. *The Psychologist* **15**, 68–71.

Montibeller G and Belton V (2006) Causal maps and the evaluation of decision options – a review. *Journal of the Operational Research Society* **57**(57), 779–791.

Montibeller G, Belton V, Ackermann F and Ensslin L (2008) Reasoning maps for decision aid: an integrated approach for problem-structuring and multi-criteria evaluation. *Journal of the Operational Research Society* **59**(5), 575–589.

O'Hagan M (1988) Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic. *Twenty-Second Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 681–689.

Olshavsky RW (1979) Task complexity and contingent processing in decision making: A replication and extension. *Organizational Behavior and Human Performance* **24**(3), 300–316.

Paté-Cornell E (1996) Uncertainties in risk analysis: Six levels of treatment. *Reliability Engenieering & System Safety* **54**(2–3), 95–111.

Pathak SD, Day JM, Nair A, Sawaya WJ and Kristal MM (2007) Complexity and adaptivity in supply networks: Building supply network theory using a complex adaptive systems perspective. *Decision Sciences* **38**(4), 547–580.

Pavlin G, Kamermans M and Scafes M (2010) Dynamic process integration framework: Toward efficient information processing in complex distributed systems. *Informatica* **34**, 477–490.

Pavlin G, Wijngaards N and Nieuwenhuis K (2009) Towards a single information space for environmental management through self-configuration of distributed information processing systems. In *Proceedings of the European Conference TOWARDS eENVIRONMENT. Opportunities of SEIS and SISE: Integrating Environmental Knowledge in Europe* (eds Hřebíček J, Hradec J, Pelikán E, Mírovský O, Pillmann W, Holoubek I and Bandholtz T). Masaryk University, Brno, pp. 94–103.

Phillips LD (1984) A theory of requisite decision models. *Acta Psychologica* **56**(1–3), 29–48.

Pratt JW (1964) Risk aversion in the small and in the large. *Econometrica* **32**(1–2), 122–136.

Raiffa H (2006) Preferences for multi-attributed alternatives. *Journal of Multi-Criteria Decision Analysis* **14**(4–6), 115–157.

Regan HM, Ben-Haim Y, Langford B, Wilson WG, Lundberg P, Andelman SJ and Burgman MA (2005) Robust decision-making under severe uncertainty for conservation management. *Ecological Applications* **15**(4), 1471–1477.

Reis S (2005) *Costs of Air Pollution Control*. Springer, Berlin.

Ríos-Insua D and French S (1991) A framework for sensitivity analysis in discrete multi-objective decision-making. *European Journal of Operational Research* **54**(2), 176–190.

Rosenhead J, Elton M and Gupta SK (1972) Robustness and optimality as criteria for strategic decisions. *Journal of the Operational Research Society* **23**(4), 413–431.

Rowe WD (1994) Understanding uncertainty. *Risk Analysis* **14**(5), 743–750.

Roy B (1991) The outranking approach and the foundations of electre methods. *Theory and Decision* **31**(1), 49–73.

Roy B (2010) Robustness in operational research and decision aiding: a multi-faceted issue. *European Journal of Operational Research* **200**(3), 629–638.

Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M and Tarantola S (2008) *Global Sensitivity Analysis: The Primer*. Wiley-Interscience, Chichester.

Schoemaker PJ and Heijden CAvd (1992) Integrating scenarios into strategic planning at Royal Dutch/Shell. *Strategy & Leadership* **20**(3), 41–46.

Schoemaker PJH (1993) Multiple scenario development: Its conceptual and behavioral foundation. *Strategic Management Journal* **14**(3), 193–213.

Schöpp W, Amann M, Cofala J, Heyes C and Klimont Z (1998) Integrated assessment of european air pollution emission control strategies. *Environmental Modelling and Software* **14**(1), 1–9.

Sigel K, Klauer B and Pahl-Wostl C (2010) Conceptualising uncertainty in environmental decision-making: The example of the EU water framework directive. *Ecological Economics* **69**(3), 502–510.

Simon HA (1979) Rational decision making in business organizations. *The American Economic Review* **69**(4), 493–513.

Siskos Y and Spyridakos A (1999) Intelligent multicriteria decision support: Overview and perspectives. *European Journal of Operational Research* **113**(2), 236–246.

Sluijs JPvd, Craye M, Funtowicz S, Kloprogge P, Ravetz J and Risbey J (2005) Combining quantitative and qualitative measures of uncertainty in model-based environmental assessment: The NUSAP System. *Risk Analysis* **25**(2), 481–492.

Snowden D (2005) Strategy in the context of uncertainty. *Handbook of Business Strategy* **6**(1), 47–54.

Stewart TJ (1992) A critical survey on the status of multiple criteria decision making theory and practice. *Omega* **20**(5–6), 569–586.

Stewart TJ (1996) Robustness of Additive Value Function Methods in MCDM. *Journal of Multi-Criteria Decision Analysis* **5**(4), 301–309.

Timmermans D and Vlek C (1992) Multi-attribute decision support and complexity: An evaluation and process analysis of aided versus unaided decision making. *Acta Psychologica* **80**(1–3), 49–65.

Torra V (1997) The weighted owa operator. *International Journal of Intelligent Systems* **12**(2), 153–166.

Valls A, Batet M and López EM (2009) Using expert's rules as background knowledge in the clusdm methodology. *European Journal of Operational Research* **195**(3), 864–875.

van der Pas JWGM, Walker WE, Marchau VAWJ, Van Wee GP and Agusdinata DB (2010) Exploratory MCDA for handling deep uncertainties: the case of intelligent speed adaptation implementation. *Journal of Multi-Criteria Decision Analysis* **17**(1–2), 1–23.

Van Dingenen R, Dentener FJ, Raes F, Krol MC, Emberson L and Cofala J (2009) The global impact of ozone on agricultural crop yields under current and future air quality legislation. *Atmospheric Environment* **43**(3), 604–618.

Van Schie E and Van Der Pligt J (1995) Influencing risk preference in decision making: The effects of framing and salience. *Organizational Behavior and Human Decision Processes* **63**(3), 264–275.

Vincke P (1999) Robust solutions and methods in decision-aid. *Journal of Multi-Criteria Decision Analysis* **8**(3), 181–187.

von Winterfeldt D and Edwards W (1986) *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge.

Wright A (2005) The role of scenarios as prospective sensemaking devices. *Management Decision* **43**(1), 87–101.

Wright G and Goodwin P (2009) Decision making and planning under low levels of predictability: Enhancing the scenario method. *International Journal of Forecasting* **25**(4), 813–825.

Xu Z (2005) An overview of methods for determining OWA weights. *International Journal of Intelligent Systems* **20**(8), 843–865.

Yager RR (1988) On ordered weighted averaging aggregation operators in multicriteria decision-making. *IEEE Transactions on Systems, Man and Cybernetics* **18**(1), 183–190.

Yager RR (2002) On the valuation of alternatives for decision-making under uncertainty. *International Journal of Intelligent Systems* **17**(7), 687–707.

Yager RR (2008) Decision making under Dempster-Shafer uncertainties. In *Classic Works of the Dempster-Shafer Theory of Belief Functions* (eds Yager R and Liu L), vol. 219 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin, pp. 619–632.

4

Preference representation with ontologies

Aida Valls[1], Antonio Moreno[1] and Joan Borràs[1, 2]
[1]*ITAKA (Intelligent Technologies for Advanced Knowledge Acquisition) Research Group, Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Spain*
[2]*Science & Technology Park for Tourism and Leisure, Spain*

4.1 Introduction

The representation and management of the user preferences is a key component in decision support systems (and particularly in recommender systems) because the solution must be based on the decision maker (i.e., the user) interests and needs. As pointed out in Adomavicius *et al*. (2010), the recommendation of some alternatives to a particular user depends usually not only on a single criterion but on a set of criteria that the user wants to take into consideration. Consequently, *multiple criteria decision making* (MCDM) methods are suitable to be applied in recommender systems, either to provide multi-criteria-based ratings of the alternatives or, in the case of collaborative systems, to correctly find other users with similar tastes to the target user.

For the sake of clarity, let us introduce here some terminology. The goal of the user (i.e., decision maker) is to select, rank or classify a set of *alternatives* (i.e., items, objects, etc.) with respect to the information provided by a set of *criteria*. Each criterion evaluates the performance of an alternative using the information provided by a reference *indicator*,

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

that is, an attribute or property observable in the alternative (indicators are usually denoted as variables in mathematics).

Three main types of approaches to *multiple criteria decision aid* (MCDA) are distinguished in the literature: functional, relational or logic models. Each one represents the user preferences in a different way.

- *Functional models*: a value system is used to associate marginal preferences upon each criterion to each of the reference indicators that describe the alternatives. These value functions permit the rating of each alternative according to its performance (Greco *et al*. 2004; Keeney and Raiffa 1976).
- *Relational models*: preferences are expressed as binary relations between the alternatives. Each criterion defines a partial preference structure $\{P, I, R\}$ on the set of alternatives, with three types of relations: preference (P), indifference (I) and incomparability (R). This type of structure is known as *outranking relations* (Perny and Roy 1992).
- *Logic models*: logic conditions are given by rules that permit the classification of the alternatives in some predetermined categories, taking into account the performance of the alternative with respect to the different criteria. In this case, the rules are part of the preference model and they are applied to either a value system or a relational preference structure (Greco *et al*. 2005).

To construct a decision preference model other additional data are needed (depending on the model) such as importance weights, substitution ratios, preference thresholds, or even a set of exemplary decisions made on a subset of the so-called *reference* actions (i.e., alternatives well known by the user). Those additional parameters heavily depend on the exploitation method that is applied to the preference data.

In this chapter we will focus on the first of the three approaches commented on above, based on the representation of the individual preferences by means of value functions. Having a set of possible values for an indicator X, and a performance scale S, a value function $g : X \to S$ associates a performance score with each value in X. These scores can be *cardinal* or *ordinal*. Cardinal scores express a known measure of the performance of a value in X, whereas ordinal information only states the relative preference position for the values on X. In both cases, *numerical* or *linguistic* values can be employed. The use of linguistic assessments permits information to be dealt with in a qualitative way. A linguistic variable takes values in a predefined set of *linguistic labels* that represent different degrees of performance, such as {None, Low, Medium, Almost High, High, Very High, Perfect}. These labels can be treated in an ordinal or cardinal way. In the first case, only their position is taken into account, so that None is worse than Low, Low is worse than Medium, etc. In the other case, the labels are mapped onto a numerical reference domain and the meaning of each linguistic label is usually given by a trapezoidal or triangular membership function (Herrera *et al*. 1996; Xu 2008; Zadeh 1975).

Concerning the indicators, the typical scales for X are *numerical* and *categorical* values. Numerical domains are defined on the real, integer or natural scales. Categorical domains are formed by a predefined and usually small set of modalities (i.e., identifiers, terms or words, such as the names of colors or of the continents). However, in the last decade, with the growth of the Semantic Web, another type of data value is receiving increasing interest, mainly in the Artificial Intelligence (AI) community: *semantic data*.

Semantic variables are those that have a large (in many cases, unlimited) set of possible linguistic/textual values, without any order or scale of measurement defined between terms. For example, the hobbies of a person can be {*Trekking, Nature photography, Italian food cooking*}. The main characteristic of this type of data value is that it can be semantically interpreted with the help of additional knowledge, such as a *domain ontology*. The analysis of these data cannot be done as if they were simple categorical modalities. The comparison between two values in a categorical variable is simply based on their equality/inequality (and sometimes is related to some kind of ordering of the categories), due to the lack of proper methods for representing the meaning of the terms. Using semantic variables it is possible to establish different degrees of similarity between values (e.g., *Trekking* is more similar to *Jogging* than to *Cooking*). Semantic similarity functions between semantic values usually depend on the ontological knowledge available for the domain of discourse (Jiang and Conrath 1997; Resnik 1995; Sánchez *et al*. 2010).

In this chapter we present a *semantic-based approach to store and exploit the personal preferences* of a user with respect to a complex domain. Recent AI knowledge models, such as ontologies, provide tools for representing the elements of a certain domain (i.e., concepts), as well as their interrelations, in a machine understandable language. They allow mapping words to concepts, so that terms can be interpreted according to their taxonomical and semantic relations (Studer *et al*. 1998). These models facilitate the design and implementation of reasoning tools that exploit the knowledge they store. A great effort has been done in some communities to develop shared domain ontologies. A paradigmatic example is the definition of shared vocabularies and thesaurus in Medicine, like SNOMED CT[1] (*Systematized Nomenclature of Medicine, Clinical Terms*), which is an ontological/terminological resource distributed as part of UMLS (*Unified Medical Language System*). It is used for indexing electronic medical records, ICU monitoring, clinical decision support, medical research studies, clinical trials, computerized physician order entry, disease surveillance, image indexing, consumer health information services, etc. Another example is the *Thesaurus on Tourism and Leisure Activities* developed by the World Tourism Organization.[2]

In this chapter we analyze around 30 *semantic-based recommender systems*, mostly published in the period 2008–2011. First, a review of different ontology-based models for preference representation is given. Secondly, we focus on how the profile can be obtained and dynamically updated. Since ontologies may have a large and complex set of concepts, it is not straightforward how to include the user preference information in this model. Some authors have proposed different mechanisms for learning the user profile, mainly based on spreading algorithms that exploit the taxonomical structure of the ontology to propagate the information about the user preferences through the whole set of concepts. The second part of the chapter is devoted to a review of these learning procedures. Thirdly, we show how different multiple criteria decision aiding tools have been used with ontology-based user profiles in recommender systems. The chapter concludes with a final summary and a discussion of the main future trends in this field.

[1] http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html (last accessed on May 9, 2012).
[2] http://www.unwto.org (last accessed on May 9, 2012).

4.2 Ontology-based preference models

Ontology-based intelligent systems have powerful modeling and reasoning capabilities. The use of explicit domain knowledge, represented in the form of an *ontology*, permits a high degree of knowledge sharing, logic inference and knowledge reuse (Wang *et al*. 2004). These knowledge structures basically describe the main concepts (and the relationships between them) in a particular domain, along with their properties and restrictions on their use, thus giving a precise meaning to each concept. Ontologies have several components on which intelligent systems may apply reasoning procedures. The main ones are classes, instances, properties and rules. A brief explanation of these features and how they are used in some ontology-based systems is given in the following:

- **Classes** are the abstract representation of the different concepts of a domain. They usually correspond with the nouns found in the domain. For instance, a class could be 'city', 'accommodation' or 'singer'. Each class has a certain number of features, represented with *slots*. For instance, the 'singer' class could have slots identifying aspects like the birth place of the singer, his/her birth date, his/her number of Grammy awards, etc.
- **Instances** of a class represent specific individuals that belong to that class of objects. For example, in the Music domain we may have instances of the class 'singer' like 'Elton John' or 'Madonna'. In the Tourism domain, 'Berlin' and 'The Plaza Hotel' are instances of the classes 'city' and 'accommodation', respectively. Instances have a particular value associated with each of their slots, including those slots inherited from all of their superclasses.
- **Properties** permit to establish binary semantic relationships between classes. The most common is the '*is-a*' property which indicates that a class is subclass of another class. For example, 'football' is-a 'sport' means that the 'football' class is a subclass of 'sport' (and, therefore, it inherits all its characteristics). This property defines a taxonomical structure of classes, which is normally a tree or an acyclic graph. Any other property between classes is considered nontaxonomical. For example, we could define the property 'locatedIn' between 'accommodation' and 'city', and use it to indicate that 'The Plaza Hotel' is located in 'Berlin'.
- Ontology **rules** are the translation of mathematical axioms that impose some constraints on the objects that can be related via a certain property or on the values that a certain slot may take. These rules may be used by ontology-based systems to implement complex reasoning mechanisms. For instance, an axiom could specify that a certain binary relationship P between classes has the transitive property; then, if the system knows that $a\ P\ b$ and $b\ P\ c$, it can infer that $a\ P\ c$.

As an example, Table 4.1 shows the main features of an illustrative set of ontologies from different domains used by ontology-based recommender systems. All these ontologies have been designed and built ad-hoc for a particular system. All the approaches use the '*is-a*' relationship (or its equivalent form 'subClass') in order to categorize the main domain concepts in a taxonomical hierarchy. Most of them also use more complex nontaxonomical relationships. For instance, Cantador (2008) defines an ontology about news that includes metadata elements like 'Subject Qualifier', 'Media Type', and 'Gender'. The Tourism ontology defined in Mínguez *et al*. (2010) has properties like

Table 4.1 List of ontologies and their main features.

Ontology	Domain	Properties (examples)	Inst.	Rules
IPTC ontology (Cantador 2008)	News	is-a, SubjectQualifier, MediaType, Gender	Yes	No
Garcia *et al*. (2011)	Tourism	is-a	Yes	No
e-Tourism (García-Crespo *et al*. 2009)	Tourism	is-a, locatedIn, interestedIn, hasCurrency	Yes	Yes
OntoMOVE (Bhatt *et al*. 2009)	Medicine	SubClass, EquivalentClass, DisjointWith, SameIndividual, DifferentFrom	Yes	Yes
Middleton *et al*. (2009)	Research papers	is-a (3 levels)	Yes	No
ContOlogy (Lamsfus *et al*. 2010)	Tourism	is-a, type	No	No
CRUZAR (Mínguez *et al*. 2010)	Tourism	subClassOf, partOf, hasQuality, location, date	Yes	Yes
OntoCrawler, OntoClassifier (Yang 2010)	Scholar	is-a	No	No
Dongxing *et al*. (2011)	Documents	is_a, hasPart, hasFunction, useMaterial, hasProperty, hasFeature, has_Standard	Yes	Yes
Luberg *et al*. (2011)	Tourism	is-a	No	Yes

Inst., instances.

'partOf', 'hasQuality', 'location' or 'date'. Another example is given by Bhatt *et al*. (2009), that uses mathematical properties such as 'equivalent', 'inverse', 'transitive' or 'functional', among others. This permits description logic reasoners to exploit the ontology, deductively inferring new facts from the available knowledge. Dongxing *et al*. (2011) used properties of documents in order to define different relationships like 'hasPart', 'hasFunction', 'useMaterial', 'hasProperty', 'hasFeature' or 'hasStandard'. In this approach, two types of semantic rules are employed to describe the low-level features of customer preferences and to build an ontological knowledge base. One is used to combine preference terms and concepts. For example, the term 'RED' and the concept

'SH-FLIP-PHONE' can form a new preference concept 'SH-RED-FLIP-PHONE'. The other rule type is applied to combine two concepts such as 'F-WORD' and 'F-TEXT' that produces the new concept 'F-WORD-TEXT'. This new concept generation is based on specific relationships, such as 'is_a', 'hasFunction' or 'hasMaterial', which give a meaning to the new concept. In the e-Tourism ontology of García-Crespo *et al*. (2009) there are properties like 'locatedIn (indoor or outdoor)', 'interestedIn', 'hasCurrency', etc. This ontological knowledge permits the system to answer questions like what activities can be visited by a certain type of tourists, which is the location of interesting places and when they can be visited. This information is inferred by using ontology rules, such as '*closeOnDate(?attraction, ?date)*' that specifies that the attraction is closed on a particular date. Another approach that uses ontology rules was presented by Luberg *et al*. (2011). For instance, a rule like '*fact(? X type architecture 0.9*?N) :- fact (? X type church ?N)*' indicates that if an item belongs to the type 'church' with score N, it can also be considered of the type 'architecture' with a score $0.9N$.

In the context of a recommender system, the information contained in the ontology is normally used to represent the main features of the different alternatives that the user is considering. The domain knowledge necessary for the recommendation can be stored in a single ontology or can be organized in multiple ontologies, each one focusing on a different dimension of the problem. For instance, García-Crespo *et al*. (2009) designed an overall taxonomy in the Tourism domain to describe attractions in general categories such as 'Gothic Art', 'Museums', 'Religious Buildings', etc. The particular attractions were represented as the instances of this ontology. Another example in the field of Tourism is given in Huang and Bian (2009) who defined a different set of classes to organize the items, such as 'Attraction', 'Location', 'OpenTimes', 'AdmissionFees' and 'Activity.' In the music recommender system reported by Celma and Serra (2008), classes are used to describe the relevant features of a song, such as 'genre', 'singer', 'title', 'duration' or 'tempo'. The route planning system defined by Niaraki *et al*. (2009) uses an ontology that represents road variables, like the traffic, safety, road facilities, weather conditions and attractions, to find the optimum path in the road network.

In other cases, more than one ontology is used, such as in the Tourism recommender system shown in Ruiz-Montiel and Aldana (2009), where there is a domain ontology formed by classes that describe implicitly the properties of a service (with classes such as 'Inexpensive Service', 'Accommodation Service' or 'Charming Accommodation Service') and a separate user ontology whose classes describe personal information, such as gender, age or touristic interests. Lamsfus *et al*. (2010) presented a semantic-based digital broadcasting contextual Tourism information system. They have created a network of ontologies, called ContOlogy, which integrates 11 ontologies, 86 classes, 63 properties and 43 restrictions. These ontologies represent the information about visitors, preferences, roles, activities, environment, devices, network, motivations, location, time and Tourism objects.

All those examples shown in Table 4.1 are based on ontologies that have been built ad-hoc to be used in the recommender system. However, different organizations and committees are defining ontologies which are publicly available. As mentioned in Section 4.1, those ontologies have been defined with the agreement of a team of experts and usually cover a larger set of concepts including much more different types of taxonomical and semantic relations. From our analysis of the recent literature, we have found a few semantic recommender systems that make use of existing ontologies or vocabularies.

These are the cases of García-Crespo *et al*. (2009) using the YAGO ontology[3] and Celma and Serra (2008) with the RDF Site Summary[4] and FOAF (Friend of a Friend)[5] ontology.

The main objective of recommender systems is to predict the degree of interest of a user for an object given the user preferences and the features of the object (Montaner *et al*. 2003). The system can then provide to the user a ranked list with the alternatives that fit better with his/her preferences. Ontologies can be applied to extend the traditional text-based recommender systems with semantic domain knowledge, with the aim of improving the accuracy of the recommendations. The hierarchical organization of the concepts in ontologies permits making a representation of both the characteristics of the alternatives and the users' preferences at different levels. Then, reasoning mechanisms can be applied to propagate the information upwards or downwards in the ontology in order to make a suitable comparison of the properties of an object with the interests of a user, to compare the properties of different objects, or to compare the interests of different users. With ontologies these comparisons can be made from a semantic (i.e., conceptual) perspective, rather than from a simple terminological matching. Section 4.4 gives more details about the methods used during the recommendation stage.

Most semantic recommender systems use ontologies to represent both the information about the alternatives and the knowledge about the user preferences. In general, alternatives are represented as instances of the ontology. In some cases, each alternative is restricted to be an instance of a unique class in the ontology. In this model, each alternative is associated with a single concept, for example 'The Plaza Hotel' is an instance of the class 'Accommodation' and of no other class. However, the most common approach is that an alternative can be an instance of several disjoint classes (Borràs *et al*. 2012). Sometimes the classes allowed are only the ones in the leaves of the taxonomy (i.e. the most specific concepts; Garcia *et al*. 2011). When an alternative is associated with multiple concepts, they can also be referred to as different annotations or keywords describing the alternative. The analysis of these multiple concepts requires some kind of multi-criteria approach.

The process of associating an alternative with the classes is called Initialization. If the set of alternatives is not fixed, some process for including new instances dynamically must be defined. In some cases, it may be also interesting to define a way to reduce the number of alternatives in the system, if they can be obsolete after a certain time (e.g., via 'forgetting' rules). The initialization process can be done manually by a domain expert, who enters the information of each new alternative and instantiates it in the corresponding ontology classes. An expert criterion is used by Albadvi and Shahbazi (2009) to design and populate the ontology. Marketing managers defined the most important nodes, such as book, CD/DVD, story or comedy. Managers also defined grain nodes as a flexible way to apply multiple rules at a time by grouping similar rules together. Moreover, they defined category attributes such as price, brand, or size that are inherited from the product category. Different products were then associated with those categories.

However, as this manual process may be long, tedious and error-prone, some works have devised automatic procedures to obtain and maintain this information. For example, Celma and Serra (2008) developed a Web crawler that extracts metadata to fill up the

[3] http://www.mpi-inf.mpg.de/yago-naga/yago.
[4] http://web.resource.org.
[5] http://www.foaf-project.org.

ontology with instances of songs, artists or concerts. It also discovers automatically relationships between artists like ‘isRelatedWith’, ‘isInfluencedBy’ or ‘isFollowerOf’. In García-Crespo *et al*. (2009) the ontology is populated with a large number of instances extracted from DBpedia,[6] which contains more than 2.49 million structured items from Wikipedia.[7] Cantador (2008) analyzed 137 254 Wikipedia entries to populate 744 classes with 121 135 instances. Other approaches consider the textual information provided by documents. In this case, some natural language processing tools are needed. For example, Yang (2010) extracts information from documents using computational linguistic techniques like normalization, segmentation, stop word filtering, word stemming and TF/IDF (term frequency/inverse document frequency) calculation. Different weights are assigned to the keywords according to their level in the hierarchy. Dongxing *et al*. (2011) analyzed the frequency of the terms in documents (alternatives) to represent weighted features for each document. A similar procedure is done by Middleton *et al*. (2009), which automatically constructs clusters of papers according to their similarity, to assign them to the same concepts in the ontology.

The second use of the ontology is in the definition of the user profile. Recommender systems, as other decision support systems, need to know the preferences of the decision maker. Different ways of making use of ontologies in the user profile can be found in the works we have studied. The simplest model associates with each user a list of keywords corresponding to the names of the classes in the ontology in which the user is interested (Bhatt *et al*. 2009; Lamsfus *et al*. 2010; Ruiz-Montiel and Aldana 2009; Shoval *et al*. 2008). However, this kind of representation provides very little information to the system. A more widespread approach consists of associating a vector of features with the user. Each feature corresponds to a different concept in the ontology (i.e., a semantic category). Then, in each user’s vector a rating of each feature is stored. This numerical value indicates the degree of interest of the user with respect to the concept (Cantador 2008; Hagen *et al*. 2005; Jiang and Tan 2009; Middleton *et al*. 2009; Sendhilkumar and Geetha 2008; Sieg *et al*. 2007; Zheng 2011). This vector approach facilitates the inclusion of other types of features in the profile, such as demographic information, as in Codina and Ceccaroni (2010), Mínguez *et al*. (2010), and Garcia *et al*. (2011). Some recent works have also taken into account some measure of the credibility associated with the information stored in the profile. The rating values may be uncertain because the user gives an approximate score due to the inference mechanisms used to obtain those values (as will be explained in the next section). A confidence degree can be associated with each rating in the profile and can be used as a weighting factor in the exploitation stage (Borràs *et al*. 2012; Codina and Ceccaroni 2010).

Finally, we can also find some works that build a specific tailored ontology for each user. In Albadvi and Shahbazi (2009) a subset of concepts is selected by the user from the ontology. Those concepts are considered as the ones relevant for the recommendation. In Blanco-Fernández *et al*. (2011) the user may select a subset of the concepts and attributes of the general ontology to generate its own ‘ontology of interest’. Then a semantic network is created, whose nodes are the class instances selected in a pre-filtering phase (i.e., the alternatives). The ontology of interest is used to identify links that relate the nodes to each other. A degree of interest is associated with each node to reflect the significance of the relationship between the alternative and the user preferences.

[6] http://dbpedia.org.

[7] http://www.wikipedia.org.

In some decision aiding tools the user profile is not updated because the system is designed to solve a single problem once. In recommender systems, the framework is completely different, since usually the goal is that the user becomes a usual client of the product. Therefore, it is crucial to maintain the user profile up to date in order to provide appropriate recommendations to the same person over time. Similarly to the case of the profile of the alternatives, there are explicit and implicit mechanisms for obtaining the information from the user. The procedures for initializing and updating the ontology-based user profiles are presented in the next section.

4.3 Maintaining the user profile up to date

In order to produce personalized recommendations to the same user over time, the system has to model the user profile about the user's interests, and maintain them up to date. Feedback information is used to modify the profile when some change on the user preferences is detected. Different types of data can be studied to model the user profile, as not only the user interests on the specific domain, but also the user context (such as the user location) is relevant. This information can be collected explicitly or implicitly.

Explicit feedback is obtained by means of the direct interaction with the user. The decision maker is requested to fill in some form (giving his/her opinion on different values of the criteria or indicating his/her location) or to rate a set of alternatives. This approach gives quite precise knowledge because the data are given directly by the user. However, it is usually considered quite an intrusive way of elicitation, and many users are not keen on spending time in answering this kind of questions.

Techniques based on *implicit feedback* aim at collecting the user information and analyzing the user's behavior in the system, such as the alternatives that are selected, purchased or viewed. More sophisticated tools study the sequence of actions done by the user on a certain alternative, or even the amount of time spent with each alternative. The main advantage of these methods is that additional effort from the user is not required. However, implicit information is more uncertain than explicit information, so less confidence must be given to it when the profile is modified.

When the user profile is based on ontologies, new techniques for the adaptation of the user knowledge must be designed. This section reviews the main approaches to this question. A key point of these methods is the exploitation of the ontological taxonomical and semantic relations in the task of learning the user profile.

Table 4.2 shows some details about the semantic recommender systems that define some kind of ontology-based user profile updating mechanism. Only 16 of the 30 papers analyzed in this study consider the maintenance of the user information. Four techniques are distinguished for the initialization of the profile. In early approaches like the one presented in Sieg *et al*. (2007), the concepts of the ontology that are associated with the user profile are obtained from the analysis of the queries that the user makes to the recommender system. It is also quite common to obtain the initial description of the user by means of forms, which may contain questions about preferences and/or demographic data (Ruiz-Montiel and Aldana 2009). Demographic information may be used to infer new preferences by analyzing the relations in the ontology. Niaraki *et al*. (2009) consider both preferences and demographics (including age, gender, nationality, marital status, language, religion, socioeconomic conditions, residence location, and ethnicity).

Table 4.2 Ontology-based profile management.

Reference	Initialization			Update			Domain inference
	Queries	Form about preferences	Form about demographic	User context	Explicit	Implicit	
Sieg *et al*. (2007)	×					×	×
Wang and Kong (2007)		×			×		
Cantador (2008)	×	×	×	×	×	×	×
Sendhilkumar and Geetha (2008)	×					×	
Shoval *et al*. (2008)		×				×	
Albadvi and Shahbazi (2009)					×	×	
Jiang and Tan (2009)	×					×	×
Bhatt *et al*. (2009)	×				×		×
Middleton *et al*. (2009)						×	×
Niaraki *et al*. (2009)		×	×	×		×	
Partarakis *et al*. (2009)						×	
Ruiz-Montiel and Aldana (2009)		×	×		×		
Codina and Ceccaroni (2010)		×			×	×	×
Lamsfus *et al*. (2010)				×			×
Blanco-Fernández *et al*. (2011)		×			×		×
Borràs *et al*. (2012)		×	×	×	×	×	×

Sometimes it is claimed that requiring so much information by means of forms is not appropriate because many users will abandon the system even before starting to use it. Hybrid approaches are used to alleviate this effect. For example one may use information about the user context to infer some of the data, given that the personal characteristics determine the human behavior and the behavior determines the context, and vice versa. In Lamsfus *et al*. (2010), the system stores the user context, such as the weather, the location and the time of the day, which is implicitly gathered from the Internet or the mobile device of the user. Niaraki *et al*. (2009) propose a model that relates the user profile with the contextual information.

Since recommendation is not a one-time task, in addition to the initial construction of the user profile the recommender system must also assure that accurate recommendations will be made in the future. Therefore, we can find different techniques for updating the decision maker profile during a session. The use of implicit methods is more widespread than the one based on the explicit requirement of feedback. In Table 4.2 we can also observe that around 25% of the papers use a combination of both approaches. Explicit knowledge elicitation has been used both in profiles based on annotations and feature vectors. For the former case, Bhatt *et al*. (2009) proposed an incremental procedure to allow the experts to refine the semantic categorization stored in the system. For the latter one, Wang and Kong (2007) used explicit information of the decision maker to update the user's degree of interest on the concepts. The user has to rate the recommended alternatives and then the degree of interest on the related concepts is modified according to the given ratings.

Several papers exploit the implicit information of the user by tracking the user's behavior. For instance, Sieg *et al*. (2007) increment or decrement the preference weights based on bookmarking, frequency of visits and time spent on each alternative (a Web page, in this case). Shoval *et al*. (2008) update the importance score of each concept based on the number of its 'clicks' divided by the total number of 'clicks' of the user. Jiang and Tan (2009) presented a method based on probabilities (Bayesian networks) for learning relations of interest. In Sendhilkumar and Geetha (2008) a weight degree is specified for each decision maker action: save (1), print (1), copy (0.25–0.75) and bookmarking (1). Those weights are applied to modify the current user profile according to the actions done on each of the proposed alternatives. A similar approach is proposed in Borràs *et al*. (2012), but they also consider the information given by the lack of actions on a certain alternative, as an indicator of noninterest.

Some approaches extend the user profile with inference mechanisms exploiting the ontology hierarchy (Codina and Ceccaroni 2010) to discover new knowledge about the user preferences. For example, if a user expresses an interest in Culture (parent class of 'Museums') it can be deduced that the user may also be interested in Museums. Conversely, if a user is interested in 'Museums', 'HumanHeritage' and 'Monuments' it could be inferred that he/she is interested in Culture in general. A derivation method for building a sub-ontology for a certain user is given by Bhatt *et al*. (2009). From the partial specification of the user's interests on a base ontology, a complete and independent sub-ontology is generated. The derivation itself is achieved by the application of different processes, like optimization schemes and consistency checking. More complex approaches like the ones of Lamsfus *et al*. (2010) or Cantador (2008) extend the user interests with spreading activation algorithms that iteratively propagate the weights of user preferences through the ontology relations. This algorithm is a method that explores networks with associated

nodes. It starts associating with a set of nodes a weight value or 'activation level' and then these weights are iteratively propagated or 'spread' to the linked nodes. The strength of the propagation normally decreases as the distance with the initial preference information increases. The process is repeated until there are no more nodes related to the initial ones. Blanco-Fernández *et al*. (2011) presented an approach in order to overcome two severe problems suffered by the traditional spreading activation algorithm. The first one is related to the kind of links that are used: some approaches have simple relationships that produce few inferences, hampering the discovery of new knowledge about complex relationships. The second problem is the propagation of static weights through the network. In order to overcome these drawbacks, they use more complex associations between nodes based on properties, such as 'hasActor', 'hasIntendedAudience', or 'isAbout'. This variety of associations permits establishing different ways to propagate the preference weights, leading to enhanced recommendations. Moreover, each semantic relation considers a different strength degree, which enables the weight to be properly updated. For instance, they consider the length of the property and the existence of a common ancestor between two nodes, among other data. Similarly, Jiang and Tan (2009) do not only consider the distance between nodes, but they also use taxonomical and join relationships. They provide a decay factor over time in the spreading process in order to represent short term preferences rather than long term ones. Borràs *et al*. (2012) proposed a spreading algorithm to propagate the user preferences on the ontology, but they also introduce an uncertainty factor associated with the interest scores. This allows modeling the confidence on the inferred values. About 60% of the reviewed papers include some domain inference mechanism.

4.4 Decision making methods exploiting the preference information stored in ontologies

Traditionally, two types of recommendation processes are distinguished (Manouselis and Costopoulou 2007): content-based and collaborative filtering. *Content-based systems* perform a matching between the features of the items to be recommended and the user interests on the values of those features. *Collaborative systems* make recommendations based on users with similar preferences. The similarity between users is normally computed by matching the rating that the users give to some of the items. Content-based systems rely on having an accurate knowledge of the user preferences to be able to select the appropriate items by simple matching. So, they suffer from the 'cold start' problem because they have poor knowledge about the user at the beginning, until the system has been able to learn the user profile properly. The weak point in collaborative systems is 'sparsity' when the number of ratings from users is small in comparison with the number of items, so that the probability of finding users that rate the same items is too low to make good estimations. Different methods to combine content and collaborative methods to build *hybrid recommender systems* have been studied (Burke 2002).

Sometimes recommendation is based not on the user's interests and preferences but on *demographic data* (e.g., age, country of origin, level of studies, etc.). This type of technique is mainly used when the system has still not enough knowledge stored in the user profile. In these cases a new user is usually assigned to a certain stereotypical class to which the people with similar demographic characteristics belong, and the system has

internal knowledge about the standard preferences of each stereotype which is used to provide the recommendations to the new user.

However, recent studies have focused on building *semantic recommender systems*, incorporating semantic knowledge to the recommendation process to minimize the main problems of traditional methods. Knowledge structures, like ontologies, are then needed to be able to make a semantic interpretation of the data values, as well as to represent taxonomical or semantic relations on the domain.

From another perspective, some recent papers agree on the suitability of taking a *multi-criteria* approach to the evaluation of the alternatives in recommender systems (Adomavicius *et al*. 2010; Shambour and Lu 2011). Two different frameworks can be devised according to the way of dealing with the information given by the user:

- *Multi-criteria preference-based user profile*: the user profile stores information that permits the evaluation of the performance of an alternative with respect to multiple descriptors associated with different properties or features of the items. Then, some exploitation method can be applied to obtain an overall rating of the alternative or to generate a ranking of the alternatives. This model is mainly used in content-based recommendation systems. The user profile can be given explicitly by the decision maker, by providing his/her general preferences on each attribute (e.g., a user prefers cheap products to expensive ones), or the user profile can be learned by analyzing the commonalities in the values of the multiple attributes of the alternatives that the user preferred (i.e., selected, viewed, bought) in the past (Marin *et al*. 2011; Marin *et al*. 2013).

- *Multi-criteria rating-based preference modeling*: this approach extends the classical collaborative filtering techniques by allowing the users to provide multiple ratings on the same item, concerning different aspects of the alternative. Instead of giving an overall rating to each alternative, the user may represent more complex preference structures by giving ratings on several attributes of the alternatives that the system shows to him/her. In this case, the system does not generalize those ratings to build a value function on the user profile, as in the previous approach. Instead, the multiple ratings are directly used to enhance the item–user and user–user comparisons. A good survey on MCDM methods in collaborative recommender systems can be found in Adomavicius *et al*. (2010).

The extension of the second type of methods with semantic knowledge has been less studied. For example, a simple method considering a two-layer taxonomy has been recently proposed by Shambour and Lu (2011), whereas Wang and Kong (2007) used the ontology to discover if two users rate items that are semantically related. Some recommender systems based on collaborative filtering use the ontological domain knowledge to compute the similarity between different users, so that they can recommend to a user those objects that were positively evaluated by users with similar tastes. For instance, a taxonomical description of hobbies could be used to infer that a user that likes Football is more similar to a user that likes Basketball than to a user that enjoys Chess. As an example, in Liang *et al*. (2009) the profile stores the tags employed by the user in a social network, which belong to a predefined taxonomy. By reasoning on the taxonomical relationships it is possible to compute the semantic similarity between users, and recommend to a user the items that similar users have tagged. Cantador and Castells

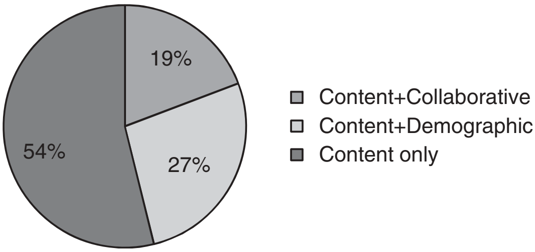

Figure 4.1 Recommendation techniques used in the reviewed works.

(2011) proposed to identify communities of interest from the tastes and preferences expressed by users in personal ontology-based profiles. A user receives advertisements about items that have been positively valued by other users in the same cluster. Collaborative filtering using ontology-based user profiles was also applied by Nocera and Ursino (2011). In this case, the authors proposed to connect user profiles creating a social folksonomy and to provide a user with a recommendation of similar users in the network. Middleton *et al*. (2009) recommended papers seen by similar people on their current topics of interest. Celma and Serra (2008), Ceccaroni *et al*. (2009) and Garcia *et al*. (2011) used the FOAF (Friend of a Friend) ontology to describe and link people.

In the rest of this section we will focus on the first approach, based on user profiles, specifically in multiple-criteria methods applied in ontology-based user profiles. Figure 4.1 shows the distribution of the types of recommendation techniques analyzed in this survey. Around half of the works that incorporate ontology-based semantic knowledge use only techniques based exclusively on the description of the content of the alternatives (54%). A significant group (27%) uses demographic information, such as predefined stereotypes in combination with the comparison of the user profile and the description of the alternatives. Only 19% of the works consider a hybrid approach that integrates both content-based and collaborative recommendation methods.

In content-based recommendation, the representations of the alternatives are compared with the representation of the user preferences. The output of the process is usually given by an overall performance score, which indicates the degree of matching between the user profile and each alternative. The higher the matching score is, the higher the performance of the alternative with respect to this particular user, because it fits well with his/her interests. In order to enable the comparison, it is assumed that a user profile and an alternative's profile share a common representation (e.g., the same set of attributes or keywords).

As explained in Section 4.3, most of the semantic recommender systems use ontologies to store the information about the alternatives. A great majority also takes advantage of the same ontology to represent the user profile. In both cases, alternatives and/or users are linked to some of the concepts in the ontology. Additionally, a rating (or weight) can be associated with these links.

A first key point in the exploitation stage is that the alternatives and the users can be compared not only by searching exactly the same term, but making use of the knowledge available in the ontology to find the degree of semantic relationship among pairs of terms. In this way, the ontology is used to bridge the gap between the terms in the users' profile and the terms representing the alternatives.

The second point concerns the method used to build a preference model associated with the alternatives. As indicated in Section 4.1, three main approaches can be distinguished: functional models, relational models and logic models. Most of the papers analyzed in this survey focus on the first case (around 80%). This approach corresponds to the model proposed in *multi-attribute utility theory* (MAUT; Keeney and Raiffa 1976), where each criterion has a value function that can be used by aggregation methods to obtain an overall performance score for the alternative. Within those works it is possible to differentiate those approaches that use some kind of *value function* (i.e., performance measure) and those that are based on computing *similarities*. Although MAUT is the more common model, some others (around 20%) design a *rule-based system* to assign a score to each alternative. In this case, a set of predefined logic rules is introduced in the recommender system (conditions → consequence). Alternatives that fulfill the conditions are assigned a certain score or category. Rules are mainly based on the equality operator and do not usually include preferential information. These three classes of approaches are analyzed in the following subsections. It may also be remarked that, to the best of our knowledge, there is not any semantic recommender system based on outranking models.

4.4.1 Recommendation based on aggregation

When the user profile is represented as a vector of features that contains the degree of interest of the user in each concept, each feature can be interpreted as a different partial criterion that can be used to evaluate an alternative. The goal is then to calculate an overall interest score for a certain alternative. The simplest approach consists of using an aggregation operator to combine the user ratings on the concepts that define a certain alternative. For instance if the alternative corresponds to a museum associated with the concepts {‘Archeology’, ‘family’, ‘Roman Empire’} the ratings of the user for these three concepts are obtained from his/her profile and are aggregated. The most usual aggregation operator is the arithmetic average (Garcia *et al*. 2011; Sendhilkumar and Geetha 2008; Yang *et al*. 2010).

If there is some additional information on the preferences, the average can be calculated with weights. For example, in Codina and Ceccaroni (2010) the confidence levels associated with the ratings are used as weights. Hagen *et al*. (2005) also considered a membership degree of the alternatives to the different categories as a weight associated with the features. In this model, the alternatives are instances of more than one class of the ontology and each instantiation has its own membership degree. Synonymous terms are also considered in the aggregation.

Some authors select an optimistic (or a pessimistic) approach to aggregate the partial ratings. This can be done by taking the maximum (or minimum) of the values (Sieg *et al*. 2007) or by using the summatory (or product) (García-Crespo *et al*. 2009). Some balance between the maximum and the minimum can be established by using the OWA (*ordered weighted averaging*) operator or the LSP (*Logic Scoring of Preferences*) method as in Moreno *et al*. (2012). In this case, for each particular application the degree of simultaneity and replaceability of the aggregation can be modeled using linguistic quantifiers (e.g., ‘most of the features are fulfilled’, ‘at least half of the features are fulfilled’).

Another possibility, as discussed in depth in Cantador (2008), is to employ the classical voting rules defined in the *social choice* field in the aggregation process. Cantador proposes the use of these techniques to find the global profile of a group of users.

Basically, two different approaches are identified: the combination of the individual preferences of the members of the group, and the combination of the ranked item lists obtained from the recommendations derived from personal profiles. In both cases, well-known voting rules such as *Borda count*, *plurality rule* or *approval voting* can be applied (García-Lapresta *et al*. 2010).

Some attempts to use a classic MCDM method, the *analytic hierarchy process* (AHP; Saaty 1980), can also be found. In general, four stages are involved in using the AHP: (1) construct a decision matrix including the value of each criterion for each alternative; (2) construct a pairwise comparison matrix of the criteria; (3) derive the relative weight of the criteria from the comparison pairwise matrix; and (4) compute the rank of each alternative based on the derived relative weight. Ontologies can be used in the first stage as in Huang and Bian (2009). In this paper the value of an alternative depends on the estimation of the preferred activities of the user, which is stored in the ontology-based user profile. In Niaraki *et al*. (2009) ontologies are used to define a family of criteria and sub-criteria. The aim is to obtain several criteria for natural disaster modeling based on an ontology-driven architecture, and to combine these criteria together in a unique function using an *analytic network process* (ANP) method (the ANP is an extension of the AHP that does not assume independence among the criteria).

4.4.2 Recommendation based on similarities

Some recommender systems annotate semantically each alternative with a subset of concepts of the ontology, which are treated as descriptive keywords. Similarly the users are also associated with a list of concepts that define the type of things they are interested in. For example in Lamsfus *et al*. (2010) the classes of the ontology define archetypes of tourists, such as cultured, sporty or adventurous. In this model, similarity measures are used to calculate the matching between the user profile and the profile of an alternative. A typical measure is the cosine similarity between the two vectors (Bhatt *et al*. 2009; Jiang and Tan 2009; Lamsfus *et al*. 2010; Sendhilkumar and Geetha 2008). A correlation measure has also been applied to measure similarity in Albadvi and Shahbazi (2009) and Middleton *et al*. (2009). Shoval *et al*. (2008) proposed a set of rules to measure the similarity between the two vectors, distinguishing *perfect match* if the same concept appears both in the user and item profiles, *close match* if the concept in the user's profile is more general than the one in the item's profile by one level (his parent) or vice versa, and *weak match* if there is a two-level difference. Ontology-based semantic similarity measures are also used in García-Crespo *et al*. (2009). The kinds of functions have been defined in the field of Computational Linguistics and permit the comparison of two terms from a conceptual point of view by exploiting the taxonomical and semantic relations represented in the ontology. García-Crespo *et al*. (2009) applied a feature-based similarity algorithm, using several ontologies as reference.

The similarity-based MCDA method TOPSIS has been also used in semantic recommender systems. It is based on the principle that the ideal solution should have the maximum similarity to the best possible solution and the minimum similarity to the worst one. The best solution would be the one with the best performance value on each criterion, and the worst solution would be the one with the worst performance value on each criterion (i.e., the combination of all the worst ratings). In Zheng (2011), the

recommendation is done on the basis of two scores of the alternatives. The first one measures the cosine similarity on the semantic annotations of the user and the alternative, whereas the second one is given by the TOPSIS method, which is used to calculate an overall utility value for each alternative with respect to its characteristics (not including the ontology concepts). The two values are aggregated with a weighted average. Another approach, proposed by Yang *et al*. (2010), filters the set of alternatives using the ontology. An expert system based on these rules is implemented (using standard inference methods). The rules compare the user profile and the description of the alternatives, based on the semantic information stored in the ontology. Then, the alternatives found with the rules are ranked using TOPSIS by analyzing their characteristics (not including the semantic information of the ontology). Notice that, in both cases, the knowledge provided by the ontology is not integrated in the TOPSIS method, but used in a separate stage of the process.

4.4.3 Recommendation based on rules

In the AI research community the design of rule-based systems for decision making has been largely studied and applied to many different domains. *Production systems* (knowledge-based systems based on rules) are based on this model of knowledge representation. Their main components include a knowledge base (composed of a set of domain-specific rules), a working memory (that contains the data of the specific problem to be solved) and an inference engine (that checks which rules are applicable to the current data and fires them). Rules define a set of logical conditions (premises) that lead to a conclusion (consequence). Rules can be evaluated using Boolean algebra or fuzzy logic (to deal with uncertainty).

Some attempts to use rules in semantic recommender systems have been found. Hsu (2009) defined a set of rules by some experts to recommend the most appropriate course to a student. The rules use the semantic annotations made on the documents associated with the different courses to classify the user in different categories. In Partarakis *et al*. (2009) a rule engine undertakes the job of selecting the appropriate interaction elements to build a personalized user interface for handicapped people. First, some rules permit the most suitable model of interface for the user to be found. Afterwards, other rules are used to adapt the interface according to the user profile, in which his/her disabilities (i.e., blind or hearing impaired) are represented.

Luberg *et al*. (2011) used RDF (Resource Description Framework) datatypes and RIF (Rule Inference Format) to define rules. When the user runs a query, the system retrieves the properties of some alternatives and sends these data to a reasoner that is able to deduce new facts about the objects by generating all the properties with confidence scores for all the items. An averaging operator is applied to obtain the overall score for each alternative.

A recommendation mechanism based on fuzzy rules has been recently proposed by García-Crespo *et al*. (2011). In this hotel recommender system the user's experience point of view is used to apply fuzzy logic techniques to relate the characteristics of the customer and of the hotel, represented by means of the domain ontology. The fuzzy rules represent the user's criterion. As the preferences may change over time, the rules are updated with the user feedback, which is received after giving the user a recommendation.

4.5 Discussion and open questions

This chapter has shown the integration of MCDA methods in the field of intelligent recommender systems. In particular, it has focused on the review and analysis of a new paradigm of recommendation tools based on semantic knowledge represented in ontologies.

In AI the design and implementation of recommender systems aims at building a tool to assist the user each time he/she needs to face a new decision on a certain domain. For example, the system may suggest a list of appropriate tourist holiday destinations in summer, or propose some films each weekend. For each recommendation the top n most interesting alternatives are selected and presented to the decision maker. Notice that an alternative should only be selected if it has not been proposed to the user before, ensuring that recommendations are new.

We can find several differences from this decision support framework with respect to the classical MCDA approach. First, the system is not designed for a unique decision maker (i.e., user) but is supposed to be used at different times by many people with different interests and characteristics. Therefore, the system must be able to manage several user profiles. Secondly, the elements of the decision making problem are not static: the list of alternatives can be different each time the user asks for a recommendation (such as the films on display each weekend), and the user profile may also change over time (e.g., preferences may vary when you get married or when you have children).

Although the design of the decision support system is different in this context than in the static MCDA problem, the recommendation of some alternatives to a particular user depends usually not only on a single criterion but on a set of criteria that the user wants to take into consideration. Consequently, MCDA methods can certainly be integrated in recommender systems, either to provide multi-criteria-based ratings of the alternatives or, in the case of collaborative systems, to correctly find other users with similar tastes to the target user.

We have reviewed the works on semantic recommendation based on content. In this case, the system selects the alternatives that fit better with the user profile. Three main approaches have been presented in Section 4.4:

(1) Alternatives are assigned an overall performance score using some aggregation operator and can then be ranked and filtered. Most of the analyzed systems are based on weighted averages. This aggregation model compensates low qualifications with high ones. This behavior is sometimes not appropriate in decision making, as argued by different authors (Bouyssou and Marchant 2007a,b; Dujmovic 2007). In the literature we can find other aggregation functions that permit the definition of different combination policies, modeling simultaneity and replacement (andness/orness), or mandatory and optional requirements, among others. For example the OWA operator or the Choquet integral have been extensively used in MCDM, but not in semantic recommender systems (Grabish and Labreuche 2010; Herrera *et al*. 1996). Another approach to minimize compensation is the one proposed in outranking methods like ELECTRE, which is based on the concept of majority and right of veto (Figueira *et al*. 2010). The application of this model to recommender systems can be seen as a new research direction.

(2) Alternatives are compared with the user profile using some similarity measure; afterwards, the most similar ones are displayed to the user. In this case the use of ontologies as the source of knowledge plays a more important role. Similarity can be based not only on exact matching of the terms, but on other kinds of information provided by the ontologies (e.g., taxonomical or semantic relations, properties of the different concepts, etc.). We observe also that semantics is poorly exploited and the few proposals that consider taxonomical relations are quite naïve. There is a large amount of literature on semantic similarity from ontologies. Three main types of measures are distinguished: edge-counting, feature-based and information content-based methods (Batet *et al.* 2011). They have been successfully applied in different domains, such as Biomedicine (Pedersen *et al*. 2007). The application of those methods in this field can be a research challenge.

(3) A set of rules, based on the user profile, are defined by the experts in order to select the most appropriate alternatives. The approach used in the reviewed systems follows the standard knowledge-based systems paradigm developed in AI, with a set of conjunctive rules and an inference mechanism to check the premises and chain the conclusions. However, in the studied works the rules are directly given by experts. When the domain is complex, the manual definition of rules may be not feasible. In this case, rules should be learned from examples. In this way, the dominance-based rough set approach (DRSA) defined for MCDA (Greco *et al*. 2002) could be an interesting model to be integrated in recommender systems. Moreover, the works on robust ordinal regression should also be considered to facilitate the modeling of the user preferences from examples (Greco *et al*. 2010).

Another key component of recommender systems is the user profile model. In this chapter we have seen how ontologies may help to represent the information about the user preferences. Some techniques for propagating the user information by means of the semantic links between concepts can be used to enhance the knowledge about the decision maker. This way of modeling the preferences could also be applied in the decision aiding process studied in MCDA. Although some of the works commented in this chapter combine some MCDA technique (such as AHP or TOPSIS) with semantic knowledge they do not fully integrate them, but use each one at different stages or in different processes. It would be much more interesting to rethink the classic MCDA methods considering the semantic view of the decision maker needs that we can have by using ontology-based user profiles.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Science and Innovation (DAMASK project, Data mining algorithms with semantic knowledge, TIN2009-11005) and the Spanish Government (PlanE, Spanish Economy and Employment Stimulation Plan). The work is also supported by the SigTur/E-Destination project funded by the FEDER European Regional Funds and the Government of the Province of Tarragona in Catalonia.

References

Adomavicius G, Manouselis N and Kwon Y (2010) Multi-criteria recommender systems. In *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners* (eds Kantor P, Ricci F, Rokach L and Shapira B). Springer, New York, pp. 769–803.

Albadvi A and Shahbazi M (2009) A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications* **36**, 11480–11488.

Batet M, Sánchez D and Valls A (2001) An ontology-based measure to compute semantic similarity in biomedicine. *Journal of Biomedical Informatics* **44**(1), 118–125.

Bhatt M, Wenny J, Prakash S and Wouters C (2009) Ontology driven semantic profiling and retrieval in medical information systems. *Web Semantics* **7**(4), 317–331.

Blanco-Fernández Y, López-Nores M, Gil-Solla A, Ramos-Cabrer M and Pazos-Arias J (2011) Exploring synergies between content-based filtering and spreading activation techniques in knowledge-based recommender systems. *Information Sciences* **181**(21), 4823–4846.

Borràs J, Valls A, Moreno A and Isern D (2012) Ontology-based management of uncertain preferences in user profiles. *14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2012)* (eds Greco S, Bouchon-Meunier B, Coletti G, Fedrizzi M, Matarazzo B and Yager RR). Springer Series: Communications in Computer and Information Science 298, 127–136.

Bouyssou D and Marchant T (2007a) An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories. *European Journal of Operational Research* **178**(1), 217–245.

Bouyssou D and Marchant T (2007b) An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories. *European Journal of Operational Research* **178**(1), 246–276.

Burke R (2002) Hybrid recommender systems: Survey and experiments. *J User Modeling and User-Adapted Interaction* **12**(4), 331–370.

Cantador I (2008) *Exploiting the Conceptual Space in Hybrid Recommender Systems: a Semantic-based Approach*. PhD thesis, Universidad Autónoma de Madrid, Madrid, Spain.

Cantador I and Castells P (2011) Extracting multilayered communities of interest from semantic user profiles: Application to group modeling and hybrid recommendations. *Computers in Human Behavior* **27**(4), 1321–1336.

Ceccaroni L, Codina V, Palau M and Pous M (2009) PaTac: Urban, ubiquitous, personalized services for citizens and tourists. In *Proceedings of the 3rd International Conference on Digital Society*, *ICDS '09*. IEEE Computer Society, Washington, DC, pp. 7–12.

Celma O and Serra X (2008) FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics* **6**(4), 250–256.

Codina V and Ceccaroni L (2010) Taking advantage of semantics in recommendation systems. In *Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence*. IOS Press, Amsterdam, pp. 163–172.

Dongxing C, Zhanjun L and Karthik R (2011) Ontology-based customer preference modeling for concept generation. *Advanced Engineering Informatics* **25**(2), 162–176.

Dujmovic JJ (2007) Continuous preference logic for system evaluation. *Continuous Preference Logic for System Evaluation* **15**(6), 1082–1099.

Figueira J, Greco S, Roy B and Slowinski R (2010) ELECTRE methods: Main features and recent developments. In *Handbook of Multicriteria Analysis* (eds Zopounidis C and Pardalos PM). Springer, Berlin, pp. 51–89.

Garcia I, Sebastia L and Onaindia E (2011) On the design of individual and group recommender systems for tourism. *Expert Systems with Applications* **38**, 7683–7692.

García-Crespo A, Chamizo J, Rivera I, Mencke M, Colomo-Palacios R and Gómez-Berbís JM (2009) SPETA: Social pervasive e-tourism advisor. *Telematics and Informatics* **26**, 306–315.

García-Crespo A, López-Cuadrado JL, Colomo-Palacios R, González-Carrasco I and Ruiz-Mezcua B (2011) Sem-Fit: A semantic based expert system to provide recommendations in the tourism domain. *Expert Systems with Applications* **38**(10), 13310–13319.

García-Lapresta JL, Marley AAJ and Martínez-Panero M (2010) Characterizing best-worst voting systems in the scoring context. *Social Choice and Welfare* **34**, 487–496.

Grabish M and Labreuche C (2010) A decade of the application of the Choquet and Sugeno integrals in multi-criteria decision aid. *Annals of Operations Research* **175**(1), 247–286.

Greco S, Matarazzo B and Slowinski R (2002) Multicriteria classification by dominance-based rough set approach. In *Handbook of Data Mining and Knowledge Discovery* (eds Kloesgen W and Zytkow J). Oxford University Press, New York, pp. 318–327.

Greco S, Matarazzo B and Slowinski R (2004) Axiomatic characterization of a general utility function and its particular cases in terms of conjoint measurement and rough-set decision rules. *European Journal of Operational Research* **158**(2), 271–292.

Greco S, Matarazzo B and Slowinski R (2005) Decision rule approach. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, New York, pp. 507–562.

Greco S, Slowinski R and Figueira J (2010) Robust ordinal regression. In *Trends in Multiple Criteria Decision Analysis* (eds Greco S, Ehrgott M and Figueira JR), Springer, New York, pp. 241–284.

Hagen K, Kramer R, Hermkes M, Schumann B and Mueller P (2005) Semantic matching and heuristic search for a dynamic tour guide. In *Information and Communication Technologies in Tourism 2005* (ed. Frew AJ). Springer, Vienna, pp. 149–159.

Herrera F, Herrera-Viedma E and Verdegay JL (1996) Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems* **79**, 175–190.

Hsu IC (2009) SXRS: An XLink-based recommender system using semantic web technologies. *Expert Systems with Applications* **36**(2), 3795–3804.

Huang Y and Bian L (2009) A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Systems with Applications* **36**(1), 933–943.

Jiang JJ and Conrath DW (1997) Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, pp. 19–33.

Jiang X and Tan A (2009) Learning and inferencing in user ontology for personalized semantic web search. *Information Sciences* **179**, 2794–2808.

Keeney RL and Raiffa H (1976) *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, Ltd, New York.

Lamsfus C, Grün C, Alzua-Sorzabal A and Werthner H (2010) Context-based matchmaking to enhance tourists' experience. *Journal for the Informatics Professional* **203**, 17–23.

Liang H, Xu Y, Li Y, Nayak R and Weng LT (2009) Personalized recommender systems integrating social tags and item taxonomy. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology – Volume 01*, *WI-IAT '09*. IEEE Computer Society, Washington, DC, pp. 540–547.

Luberg A, Tammet T and Järv P (2011) Smart city: A rule-based tourist recommendation system. In *Information and Communication Technologies in Tourism (ENTER 11)* (eds Law R, Fuchs M and Ricci F), Springer, Vienna, pp. 51–62.

Manouselis N and Costopoulou C (2007) Analysis and classification of multi-criteria recommender systems. *World Wide Web* **10**(4), 415–441.

Marin L, Isern D, Moreno A and Valls A (2013) On-line dynamic adaptation of fuzzy preferences. *Information Sciences* **220**, 5–12.

Marin L, Moreno A and Isern D (2011) Automatic learning of preferences in numeric criteria. artificial intelligence research and development. In *Frontiers in Artificial Intelligence and Applications* (eds Fernandez C, Geffner H and Manyà F), vol. 232. IOS Press, Amsterdam, pp. 120–129.

Middleton SE, De Roure C and Shadbolt NR (2009) Ontology-based recommender systems. In *Handbook on Ontologies* (eds Staab S and Studer. R), International Handbooks on Information Systems. Springer, Berlin, pp. 779–796.

Mínguez I, Berrueta D and Polo L (2010) CRUZAR: An application of semantic matchmaking to e-tourism. In *Cases on Semantic Interoperability for Information Systems Integration: Practices and Applications* (ed. Kalfoglou Y). IGI Global, Hershey, PA, pp. 255–271.

Montaner M, López B and de la Rosa JL (2003) A taxonomy of recommender agents on the internet. *Artificial Intelligence Review* **19**(3), 285–330.

Moreno A, Valls A, Isern D, Marin L and Borràs J (2012) SigTur/E-destination: Ontology-based personalized recommendation of tourism and leisure activities. *Engineering Applications of Artificial Intelligence*. http://dx.doi.org/10.1016/j.engappai.2012.02.014.

Niaraki AS, Kim K and Lee C (2009) Ontology based personalized route planning system using a multi-criteria decision making approach. *Expert Systems with Applications* **36**, 2250–2259.

Nocera A and Ursino D (2011) An approach to providing a user of a 'social folksonomy' with recommendations of similar users and potentially interesting resources. *Knowledge Based Systems* **24**(8), 1277–1296.

Partarakis N, Doulgeraki C, Leonidis A, Antona M and Stephanidis C (2009) User interface adaptation of web-based services on the semantic web. In *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments* (ed. Stephanidis C), vol. 5615 of Lecture Notes in Computer Science. Springer, Berlin, pp. 711–719.

Pedersen T, Pakhomov S, Patwardhan S and Chute C (2007) Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics* **40**(3), 288–299.

Perny P and Roy B (1992) The use of fuzzy outranking relations in preference modeling. *Fuzzy Sets and Systems* **49**, 33–53.

Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial intelligence - Volume 1*, *IJCAI'95*. Morgan Kaufmann, San Francisco, CA, pp. 448–453.

Ruiz-Montiel M and Aldana JF (2009) Semantically enhanced recommender systems. In *On the Move to Meaningful Internet Systems: OTM 2009 Workshops* (eds Meersman R, Herrero P and Dillon T), vol. 5872 of Lecture Notes in Computer Science. Springer, Berlin, pp. 604–609.

Saaty TL (1980) *The Analytic Hierarchy Process*. McGraw-Hill, New York, NY.

Sánchez D, Batet M, Valls A and Gibert K (2010) Ontology-driven web-based semantic similarity. *Journal of Intelligent Information Systems* **35**, 383–413.

Sendhilkumar S and Geetha TV (2008) Personalized ontology for web search personalization. In *Proceedings of the 1st Bangalore Annual Compute Conference*, *COMPUTE '08*. ACM, New York, NY, pp. 18:1–18:7.

Shambour Q and Lu J (2011) A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Volume 01*, *WI-IAT '11*. IEEE Computer Society, Washington, DC, pp. 71–78.

Shoval P, Maidel V and Shapira B (2008) An ontology-content-based filtering method. *International Journal of Information Theories and Applications* **15**(4), 51–63.

Sieg A, Mobasher B and Burke R (2007) Web search personalization with ontological user profiles. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, *CIKM '07*. ACM, New York, NY, pp. 525–534.

Studer R, Benjamins VR and Fensel D (1998) Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* **25**(1-2), 161–197.

Wang RQ and Kong FS (2007) Semantic-enhanced personalized recommender system. In *2007 International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 4069–4074.

Wang XH, Zhang DQ, Gu T and Pung HK (2004) Ontology based context modeling and reasoning using OWL. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, *PERCOMW '04*. IEEE Computer Society, Washington, DC, pp. 18–22.

Xu Z (2008) Linguistic aggregation operators: An overview. In *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models* (eds Bustince H, Herrera F and Montero J), vol. 220 of Studies in Fuzziness and Soft Computing. Springer, Berlin, pp. 163–181.

Yang L, Hu Z and Long J (2010) Service of searching and ranking in a semantic-based expert information system. In *Proceedings of the 2010 IEEE Asia-Pacific Services Computing Conference*, *APSCC '10*. IEEE Computer Society, Washington, DC, pp. 609–614.

Yang S (2010) Developing an ontology-supported information integration and recommendation system for scholars. *Expert Systems with Applications* **37**(10), 7065–7079.

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning – I. *Information Sciences* **8**, 199–249.

Zheng Z (2011) A personalized product recommender system based on semantic similarity and topsis method. *Journal of Convergence Information Technology* **6**(7), 312–320.

Part III

DECISION MODELS

5

Neural networks in multicriteria decision support

Thomas Hanne
University of Applied Sciences and Arts Northwestern Switzerland, Switzerland

5.1 Introduction

Originally, artificial neurons and neural networks were simplified models of nerve cells and systems used to explain biological information processing (McCulloch and Pitts 1943). Later on and particularly since the 1980s numerous other applications have been found, many of them without any direct relationship to the biological paradigm. Since then neural networks (sometimes denoted as artificial neural networks) have been considered as an appealing means for solving various kinds of problems in engineering, economics, computer science, etc. Based on the intriguing biological analogy of information processing in nerve nets, they have attracted researchers from many disciplines for further theoretical studies, methodological developments and practical applications. The applications mainly aim at technical areas and at those fields where artificial intelligence (AI) is required but where traditional logic and symbol based approaches often fail, e.g. in classification, prediction, or pattern recognition.

The main idea of neural networks is to use simple connected units (the neurons) as computational models. These units work in parallel and may show together a rather complex functionality. In fact, for some types of neural networks it could be proven that they allow to calculate any Boolean function or to approximate any kind of continuous function (see below for further details).

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

The other appealing feature of neural networks is that they usually employ some learning algorithm which allows to specify them in some automatic way for fulfilling the desired purposes. Various algorithms for machine learning have been suggested to specify the design or the parameters of a neural network. Some of these approaches are denoted as supervised learning as they require training data together with desired outputs. Other forms denoted as unsupervised learning only require training data without a prespecified desired output. Typical applications of the latter concept are in the area of clustering with the purpose of finding patterns or similarities in the input data.

Since nerve nets (although being much larger than typical neural networks) are the basis for human intelligence it appears to be natural to consider neural networks as a suitable tool for supporting human decision making. Focusing on the fact that multiple criteria or objectives are almost always relevant in human decision making we explore the suitability of neural networks for this particular type of problem in the subsequent sections. In Section 5.2 we start with a review of basic neural network concepts. In Section 5.3 we briefly reconsider tasks in multiple criteria decision support. Section 5.4 is devoted to the discussion of application scenarios for neural networks in multicriteria decision support. The chapter ends with a summary and an outlook in Section 5.5.

5.2 Basic concepts of neural networks

Numerous types of neural networks have been suggested in the literature (see e.g., Hecht-Nielsen Hecht-Nielsen 1987; Müller and Reinhardt 1990; Rojas 1993). Their common idea is that there is a number of simple and uniform information processing units, the neurons, which are connected in some way and exchange information accordingly. From an abstract point of view a neural network N can be described as a finite directed graph where the nodes are the neurons (and possibly, additional input nodes) and the edges correspond to the connections. Each edge is valued by a weight which corresponds to the strength of forwarding information from one neuron to another neuron. Formally written, the neural network can then be described by $N = (V, E, I, O, w, F)$ with the following properties:

(i) $(V \cup I, E)$ is a finite directed graph with vertices (nodes) $V \neq \emptyset$ and edges $E \subseteq (V \cup I) \times V$;

(ii) $I \cap V = \emptyset$, I (set of input nodes) is finite;

(iii) $O \subseteq V$ (set of output nodes) holds;

(iv) $w : E \to R$ are the weights assigned to edges;

(v) $F = \{F_v : R^{|V(v)|} \to R : v \in V\}$ is a set of node-specific neuron functions.

For simplicity, the nodes of V can be denoted by natural numbers $\{1, \dots, n\}$ with $n := |V|$ being the number of nodes. A node $i \in V$ is called a *neuron* whereas I is called the set of *input nodes* and O the set of *output nodes*. Let $o = |O|$ be the number of output nodes. For $(i, j) \in E$, $w_{ij} = w(i, j)$ is called the *weight* or *connection strength* of i to j. For $i \in V$, $F_i = F(i)$ is called the *function* of neuron i. For $i \in V \cup I$, $N(i) := \{j \in V : (i, j) \in E\}$ is the set of succeeding neurons numbered as $N(i) = \{n_{i1}, \dots, n_{i|N(i)|}\}$.

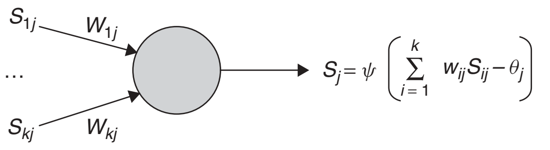

Figure 5.1 Function calculated by a single neuron j.

For $i \in V$, $V(i) := \{j \in V \cup E : (j, i) \in E\}$ is the set of preceding neurons or input nodes numbered as $V(i) = \{v_{1i}, \ldots, v_{|V(i)|i}\}$.

Depending on its input value the output of a neuron can be described in most models as follows: the *state (output)* S_j of a neuron j is calculated by a function F_j of the states of the preceding neurons (Figure 5.1):

$$S_j = F_j(S_{1j}, \ldots, S_{|V(j)|j}) = \psi\left(\sum_{i \in V(j)} w_{ij} \cdot S_i - \theta_j\right). \tag{5.1}$$

In Equation (5.1), the S_{ij} are the states or input values of the preceding neurons or input nodes, respectively. These input values are multiplied by weights w_{ij} indicating the connection strength to the considered neuron. The weights $w_{ij} \in [-1, 1]$ specify in which proportion the output of the i th neuron (or input) is considered as exciting ($w_{ij} > 0$) or inhibitive ($w_{ij} < 0$) with respect to the output value of the jth neuron. From the sum of input values multiplied by the respective weights, θ_j, a threshold value, is subtracted.

Using a function ψ the result is mapped into a bounded domain such as the interval $[0, 1]$ or $[-1, 1]$. ψ is considered to be an increasing function defined on the domain of real numbers which fulfills $lim_{x\to\infty}\psi(x) = 1$ and, depending on the network model, $lim_{x\to-\infty}\psi(x) = 0$ or -1. Such a bounded, strictly increasing function on R is usually called a sigmoid function if it is differentiable (Hecht-Nielsen 1987, p. 106).

In a rather generic way, the functionality of a neuron can be defined (Rojas 1993, p. 31) as

$$S_j = \psi_j(\phi_j(S_{1j}, \ldots, S_{|V(j)|j})) \text{ or } F_j = \psi_j \circ \phi_j, \tag{5.2}$$

where the ϕ_j may as well be other functions than the above weighted sum minus a threshold value. The inner function ϕ_j is also called an *activation function* while ψ_j is also referred to as an *output function* or *transfer function*. The total state of a network can be written as a vector S. As result of a neural network we can consider the vector of states of output nodes.

The first approach for a formal modeling of the working of neurons as logical threshold elements is due to McCulloch and Pitts (1943) who define ψ as a step function:

$$\psi(x) = \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases} \tag{5.3}$$

For the weights of two connected neurons i, j, it is assumed that $w_{ij} \in \{-1, 1\}$ holds.

In stochastic neuron models one usually chooses for ψ the so-called Fermi function

$$\psi(x) = (1 + e^{-2\beta x})^{-1} \tag{5.4}$$

which can be interpreted as the probability of an output 1 or understood as an output on a continuous domain [0, 1]. The term 'Fermi function' refers to a thermodynamical analogy as it describes the thermal energy of a system of identical fermions. Parameter $\beta > 0$ is defined as an inverse temperature $T = 1/\beta$. The analogy allows the application of methods from statistical physics for analyzing neural networks in more detail (Müller and Reinhardt 1990, p. 37). For $\beta \to \infty$, ψ converges to the step function (3).

Using a transformation $\psi \mapsto 2\psi - 1$ of the interval [0, 1] to the interval $[-1, 1]$ one obtains the output function

$$\psi(x) = \tanh(\beta x) = \frac{1 - e^{-2\beta x}}{1 + e^{-2\beta x}} \tag{5.5}$$

which is also frequently used in neural network models.

Another model, the linear associator, assumes a linear increase of the output states of 0 (or -1) to 1 within a given interval (Brause 1991, p. 41), Rosenblatt (1958, 1962) proposes a type of neural networks, called (simple) perceptron where the neurons are arranged in one layer. The input values build up a preceding layer with given connections to the neurons.[1] Between the inputs and the neurons, there is a 'complete connection structure' while the neurons are not connected to each other. The number of output values corresponds to the number of neurons. For the weights w_{ij} which, in this case, denote the weighting of the ith input at the jth neuron are assumed to be either $+1$ or -1, i.e., $w_{ij} \in \{-1, 1\}$.

A typical and presumably most popular type of connection structure is the (layered) feedforward layered networks where the neurons are organized in layers, as shown in Figure 5.2. Neurons have only connections with other neurons from the preceding and the succeeding layer. This approach can be considered as perceptron networks which are generalized for two or more layers (multilayer perceptrons). Signals (information) are

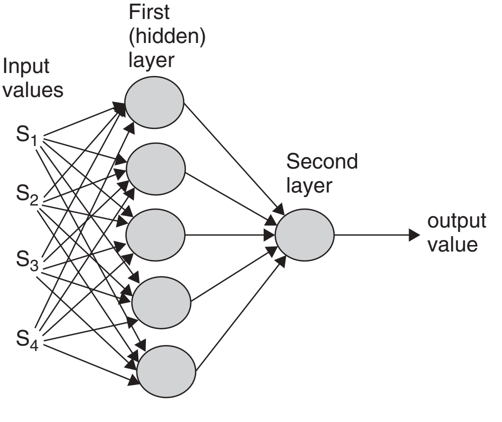

Figure 5.2 Feedforward neural network with 4 input values, one hidden layer of 5 neurons, and 1 output neuron.

[1] Sometimes this layer is counted as an additional layer when dealing with the perceptron, e.g., in Müller and Reinhardt (1990). Often in the literature, e.g., in Ritter *et al*. (1991), the input data are not considered as a layer. This also holds within the framework of this chapter.

only conducted from neurons of one layer to those of the subsequent layer. Between such neighboring layers there is a complete connection structure, i.e., each neuron of the preceding layer is connected to each neuron of the subsequent layer. There are no feedbacks. The additional layers preceding the output layer are also denoted as hidden layers. The input neurons receive their inputs from the input nodes, the others receive their inputs as outputs of the preceding neurons. More formally, such a feedforward network consisting of s layers can be defined as follows: $V = V_1 \cup \ldots \cup V_s, V_i \neq \emptyset, V_i \cap V_j = \emptyset$ for all $i, j \in \{1, \ldots, s\}, i \neq j, O = V_s$. With $V_0 := I$, E can be defined as follows: $E := \{(i, j) : i \in V_k, j \in V_{k+1}, k \in \{0, \ldots, s-1\}\}$.

A preliminary consideration for utilizing neural networks is to analyze their theoretical power. Neural networks are not universal models of calculation as, for instance, Turing machines or abstract von Neumann computers. Minsky (1967) (see also Rojas 1993, p. 47) showed the equivalence of (recursive) McCulloch-Pitts networks with the class of finite automata which are much less powerful models of computation than, e.g., Turing machines. Furthermore, the equivalence of McCulloch-Pitts networks with absolute inhibition, McCulloch-Pitts networks with relative inhibition (Rojas 1993, p. 42), and networks with rational weights (Rojas 1993, p. 40) can be proven.

Using a two-layered McCulloch-Pitts network, it is possible to calculate any logical function $\rho : \{0, 1\}^n \rightarrow \{0, 1\}$ (see e.g., Rojas 1993, p. 39; Muller and Reinhardt 1990, p. 55). Thus, the McCulloch-Pitts threshold elements build up a complete logic as, e.g., AND and NOT building blocks. Lapedes and Farber (1988) proove that feedforward networks with 'Fermi neurons' can approximate arbitrary multidimensional continuous functions. For one-dimensional functions, a neural network with one hidden layer is sufficient, for multidimensional functions two hidden layers are required. Hecht-Nielsen (1987) shows that generally one hidden layer is sufficient. A more rigorous proof which assumes typical output functions ψ as often used in feedforward networks is due to Hornik *et al.* (1989).

These theoretical results possibly can ensure that neural networks with certain desired properties do exist. However, this does not guarantee that one actually can determine them easily. Neural networks can be constructed for a specific purpose 'manually'. But frequently, it is not clear how this should be accomplished. The presumably most important feature of neural networks from this point of view comes into operation, namely the provision of a desired functionality by training a neural network (training phase) prior to validating (test phase) and applying it (working phase). During the training phase, learning and a storage of information is done by a suitable modification of parameters of the network, e.g., the connection strength or weights. For doing so, e.g., historically, typical input data together with the corresponding reference output data can be used to provide a descriptive definition of the desired functionality. For realizing such a learning behavior, various methods are proposed in the literature which are mostly tailored to a specific type of neural networks.

Typically, the learning concepts are differentiated into supervised and unsupervised learning. For the supervised learning, usually starting with randomly determined weights, these are improved during the training phase. For doing so, training inputs and corresponding reference outputs which indicate the desired output of the neural network are used. The error of the network (mean absolute error) defined as the distance of the outputs calculated for the training inputs to the reference outputs shall be minimized. During the unsupervised learning there are no reference outputs, i.e., no specific network functions.

Instead, there is a self-organization of the network which can be used for finding patterns or similarities in the input data (e.g., for clustering applications). The most often used type of this network is called a Kohonen network or self-organizing map (see e.g., Kohonen 1982; Rojas 1993, pp. 339–359).

Supervised learning dominates in the majority of applications. This approach can be subdivided into adjusting learning which is based on the determination of suitable parameter values in one step and reinforced learning which uses a stepwise adaptation of weights and other parameters, e.g., threshold values, where applicable. Let us also remark here that for the learning mostly a given type of network and a fixed topology and size of the network is assumed.

The most well-known learning method for neural networks of the feedforward type is the error backpropagation method (see, e.g., Müller and Reinhardt 1990, pp. 46–55) which follows the basic ideas of gradient-based optimization. This method was originally suggested for the perceptron neural network. In the simple perceptron without hidden layers a gradient approach is used to assess a modification of the weight w_{ij} (which is a measure for the connection strength from the ith neuron or input to the jth neuron or output):

$$\Delta w_{ij} = \epsilon \sum_{\mu=1}^{z} (\zeta_j^{\mu} - S_j^{\mu}) e_i^{\mu}.$$

Here, $0 < \epsilon \ll 1$ is a given parameter, the $e_i^{\mu}, \mu \in \{1, \ldots, z\}$ are z input values, and the $S_j^{\mu}, \mu \in \{1, \ldots, z\}$ are the corresponding output values. The $\zeta_j^{\mu}, \mu \in \{1, \ldots, z\}$ are the desired output values of the jth neuron. We start with given, possibly randomly determined, weights and a sample of input and corresponding desired output values. Then we calculate the actual output values, tune the weights with the help of the above learning rule, and repeat this procedure with the same or new data samples until satisfactory results are achieved. A generalization of this concept for multilayer networks leads to the error backpropagation.

The problem of choice of network structure and size is usually left out of consideration in this and similar learning concepts. For the problem of determining the size and structure there are relatively few theoretical analyses (see e.g., Baum and Haussler 1989). The methods developed for this purpose are often pure heuristics whose effectiveness or efficiency can hardly be proven.

As a learning concept which is independent of the type of network and able to support an adaptation of the network structure, evolutionary algorithms have become popular during the last 10–15 years (see e.g., Sexton *et al*. 2003). Evolutionary algorithms are based on the idea of using concepts from natural evolution for finding good solutions to hard optimization problems. This type of metaheuristic search strategy became popular in many other application areas apart from the application in neural network construction and learning. Of course, other optimization techniques and methods from the field of metaheuristics could be used in neural network applications as well but are, in fact, seldom discussed in related research.

Let us briefly consider the basic steps of evolution strategies (Schwefel 1981) as one representative of evolutionary algorithms which can be used for neural network learning (Hanne 1997). Evolution strategies simulate the main evolutionary principles which are replication, mutation, recombination, and selection. For learning in a neural network first a population of networks need to be built, e.g. randomly. This population is to be

improved during a number of generations. In each generation, first networks from the parental generation which consists of $\mu, \mu \in N$, entities, are replicated. The replication leads to $\lambda, \lambda \in N$, children networks with data that are mutated, i.e., children networks need to differ at least slightly from their parent networks. It is also useful to recombine the data so that a child network inherits its information from different parent networks. Then the fitness values of the children methods are calculated and the μ best are selected as parents for the next generation. In the $(\mu + \lambda)$-evolution strategy children and parents are considered in this selection process whereas in (μ, λ)-evolution strategies only the children can survive.

The calculation of the fitness of a method is performed similar to the perceptron learning rule approach, i.e., a neural network is applied to some training input and the result is compared with a given reference result. The distance measured by means of an l_p metric between the calculated result and the desired result is to be minimized during the evolutionary process.

5.2.1 Neural networks for intelligent decision support

Neural networks are one of the often used methods in the area of AI research. Traditional approaches in the area of expert systems are based on an observable and articulable process of logical reasoning. Such processes can be considered as the most sophisticated way of human thinking but it is a rather novel, slow and error-prone way of biological information processing. In contrast to that, neural networks correspond to the silent and unobservable processes of human information processing. For instance, the processing of sensory data, pattern recognition, and the memory activities are among the most powerful and mainly used methods of human information processing for which there is no verbal protocol, which makes these activities transparent. In fact, traditional expert systems have difficulty in accomplishing the mentioned tasks which are typical areas of applying neural networks – often with significant success.

Neural networks are, hence, an important alternative or supplement to traditional expert systems. A main difference to expert systems based on the paradigm of symbol processing (Newell 1981) is that information is not stored in some compact form (symbolic representation) which could easily be interpreted by a human. Instead, all information and thus the functionality of the neural network is stored in some distributed way – in the connection strengths between the neurons and possibly other parameters. Because of the lack of symbolic representation, neural networks are often regarded as ‘black boxes’ (Rojas 1993, p. 29). Thus, it is difficult to understand or to explain what neural networks actually do, why a neural network has a specific parameter setting and why the output is calculated in some specific way.

In contrast to traditional expert systems the working of a neural network cannot easily be reconstructed using human reasoning. The output is simply the result of all input data and the information stored in the network (e.g., in the connection strengths). This fact is one of the most important drawbacks compared with conventional expert systems but at the same time builds the basis for many desirable properties of neural networks. On the other hand, it is advantageous for neural networks that there is a possibility of automated learning while for expert systems often the knowledge base has to be constructed manually in a time consuming and costly way.

Because of their information redundancy, neural networks have also an advantageous error tolerance, stability, and the capability to generalize: similar input data usually result in similar output data. From the learning based on a relatively small training set, the network can conclude to the input–output relationships of further unknown data. Compared with traditional expert systems, these features let neural networks appear as an often advantageous, intelligent decision instrument capable of learning. The error tolerance allows for dealing with perturbed, incomplete or inconsistent data. This is of special importance if, e.g., information provided by a decision maker shall be processed but there are uncertainties about their specific values. Neural networks allow for dealing with ill-structured problems and the modeling of nonlinear relationships. An important application is, for instance, the prediction of nonlinear and perturbed time series (Pau and Jones 1994).

Neural networks have already shown an impressive spectrum of successful applications. Among them, possibilities of application predominate in the area of engineering where technical data are to be analyzed, e.g., the processing of sensory information such as vision or speech recognition. Attention has also increasingly been directed to economic applications. In this connection, neural networks have proven to be a promising instrument also for analyzing complex socio-economic systems and macroeconomic variables (Herbrich *et al*. 1999). In doing so, it is basically also possible to process qualitative data represented by numbers and to interpret the output of a neural network qualitatively.

The possibility of applying neural networks to decision support is analyzed, e.g., by Schocken and Ariav (1994) and Tam (1994) who discuss the following key applications: classification, pattern recognition, multiattribute scoring and prediction (see e.g., Caporaletti *et al*. 1994). Especially in the areas where large multitudes of quantitative information arise, and these are typically the financial markets, successful applications have been presented. For instance, neural networks, especially feedforward networks, have been used for stock price prediction, interest rate prediction, the prediction of currency rates, of bankruptcies of industrial or banking companies, of purchasing patterns and bond payoffs. For further information, see e.g., Dutta and Shekhar (1988) White (1988), Kamijo and Tanigawa (1990), Kimoto *et al*. (1993), Yoon and Swales (1993), Dutta *et al*. (1994), Piramuthu *et al*. (1994), Tam (1994), Wilson (1994) and Yoon *et al.* (1994).

The recommended decisions derived from such predictions often appear to be profitable and superior to those of other, more traditional methods (chart techniques, regression methods, discriminant analyses, etc.). Hill and Remus (1994) generally propose to apply neural networks for management decisions and justify this with respect to often used simple linear decision rules (e.g., regression models). Compared with these, neural networks often have proven to be superior. Sharda (1994) supports the integration of OR and neural network research by providing a bibliography for OR researchers. Yoon *et al*. (1994) study the integration of feedforward networks with classical rule-based AI systems.

An interesting application of feedforward networks is also presented by Chu and Widjaja (1994) who approach the (meta decision) problem of determining a prediction method. For the considered problems with short, perturbed time series, neural networks themselves appear hardly suitable as prediction methods. Instead, it seems to be purposeful to apply feedforward networks here for selecting a method or for combining the prediction results of different methods. A similar approach is due to Sohl and Venkatachalam (1995).

5.3 Basics in multicriteria decision aid

5.3.1 MCDM problems

In multicriteria decision aid (MCDA) [often also denoted by multiple criteria decision making (MCDM) or multicriteria decision support] decision problems are analyzed for which several criteria or objectives should be taken into account at the same time. Such a problem can formally be defined as follows: let $A \neq \emptyset$ be a set of *alternatives* (also called *actions* or *strategies* or *solutions*) of a decision problem. Let

$$f : A \to R^q \tag{5.6}$$

be a multicriteria evaluation function. A proper MCDM problem is given only for $q \geq 2$ criteria or objectives. However, the case of an ordinary scalar optimization problem with $q = 1$ can be considered as a special case of an MCDA problem. Each function $f_k : A \to R$ with $f_k(a) = z_k \quad (k \in \{1, \ldots, q\}, a \in A)$ with $f(a) = (z_1, \ldots, z_q)$ is called a *criterion* or *objective function* or *attribute*. Let us assume that each criterion is to be maximized, thus that a higher value in each criterion is always preferred to a smaller value.[2] (A, f) is called an MCDA problem.

Traditionally, the MCDM literature (Hwang and Masud 1979; Hwang and Yoon 1981) distinguishes two main cases of such problems: MADM (multiple attribute decision making) and MODM (multiple objective decision making). In MADM it is assumed that the decision problem has an explicitly given finite (and usually rather small) set of alternatives. MODM is based on the assumption of a large and often infinite set of alternatives which is usually specified as a subset of a vector space defined by restrictions. While the criteria values of the alternatives are explicitly stated for a MADM problem, objective values in MODM are given implicitly in the form of objective functions.

Formally written, an MCDM problem can be specified as

$$\mathcal{P} = (A, f)$$

and is also called a *MADM problem* if A is finite. In this case, problem $\mathcal{P}$ can be described by a MADM *decision matrix* $Z \in R^{l \cdot q}$ where

$$A = \{a_1, \ldots, a_l\} \tag{5.7}$$

and

$$a_h = (z_{h1}, \ldots, z_{hq}) \tag{5.8}$$

for all $h \in \{1, \ldots, l\}$ holds.

An MCDM problem $\mathcal{P} = (A, f)$ is called a *MODM problem* if A can be written as

$$A \subseteq R^n, A = \{a \in R^n : g_i(a) \leqq 0, i \in \{1, \ldots, m\}\} \tag{5.9}$$

for functions (restrictions)

$$g_i : R^n \to R, i \in \{1, \ldots, m\}. \tag{5.10}$$

[2] If a criterion $f_k^{'}$ has to be minimized, then a substitute maximizing criterion can be defined by $f_k := -f_k^{'}$.

Multiple objective linear programming (MOLP) is an important special case of MODM with linear objective functions and linear restrictions. Such kind of problem can be considered as a generalization of linear programming which has been extensively treated in the literature (see, e.g., Gal 1995, pp. 335–364 and Steuer 1986). Other important special cases and extensions of MODM problems are, for instance, in the areas of combinatorial and discrete optimization, stochastic optimization, and fuzzy optimization.

The basic idea in MCDM is that in many real-life decision problems several criteria are to be considered at the same time. This aspect is often neglected in traditional approaches which simply replaced the multiple criteria by *one* hypothetical or constructed general criterion for choice. For instance, economic theory is, for a large part, built on the concept of utility which is assumed to be the objective to be maximized by individuals or households demanding commodities. Another simple and traditional approach is to aggregate the criteria using a weighted sum, e.g., in engineering applications. Other ad-hoc approaches of aggregating the multiple criteria to a single objective are in use as well. In current MCDM research, more attention is being given to the treatment of several objectives, the development of new methods, and the application of formal methods in practice.

5.3.2 Solutions of MCDM problems

When comparing vectors $x, y \in R^q$ in objective space, the following three relations are useful:[3]

$$x \leqq y \quad \text{iff} \quad x_k \leqq y_k \quad \forall k \in \{1, \ldots, q\}, \tag{5.11}$$

$$x \leq y \quad \text{iff} \quad x \leqq y \quad \textit{and} \quad x \neq y, \tag{5.12}$$

$$x < y \quad \text{iff} \quad x_k < y_k \quad \forall k \in \{1, \ldots, q\}. \tag{5.13}$$

When taking several objectives into account for a decision problem, there is usually not an *optimal* solution which is better than all other solutions with respect to each of the objectives. Instead, we can require for a solution just that it is not possible to improve one criterion without deteriorating at least one other criterion. This consideration leads to the concept of what is usually called a *Pareto-optimal* or *efficient* solution. For a multicriteria maximization problem the *efficient set* $E(A, f)$ is defined as

$$E(A, f) := \{a \in A : \neg\exists b \in A : f(a) \leq f(b)\}, \tag{5.14}$$

where each $a \in E(A, f)$ is called *efficient*. Note that there is usually not a unique efficient solution but a set of solutions with different evaluations concerning the objective functions.

Occasionally, also the image of $E(A, f)$ in criterion space defined by

$$f(E(A, f)) = \{f(a) : a \in E(A, f)\} \tag{5.15}$$

is denoted as an efficient set. For $a, b \in A$, we say that *a dominates b* (with respect to f); in symbols: $a \succ_f b$ or just $a \succ b$ (if definite), if $f(b) \leq f(a)$. $b \notin E(A, f)$ is also called *inferior, dominated* or *inefficient*.

[3] Occasionally the symbol $\leqq$ is used instead of $\leq$ for scalar values in order to treat them as a special case of vectors.

From an application-oriented viewpoint the determination of all efficient solutions is not useful because the decision maker usually just wants to select one alternative as the solution of a real-life decision problem. The case of a single efficient solution which dominates all other alternatives is a special case (*perfect solution*) such that no true MCDM problem is given. Frequently, even in the case of a finite number of alternatives, the efficient set is so large that the decision maker can hardly examine each of them. The property of efficiency can, however, serve as the basis of a filtering method for reducing the considered set of alternatives (because usually only efficient alternatives can be considered as rational candidates for choice). On the other hand, it should be mentioned that, for instance, too strong simplifications and errors in the formalization of a decision problem may have the consequence that 'inefficient' alternatives could definitely be relevant (see e.g., Zeleny 1982, pp. 142–148).

Compared with a pure determination of the efficient set, application-oriented methods usually require that additional information or assumptions, e.g., information on the preferences of a decision maker, are provided for the method to determine one or a few alternatives which are sometimes called compromise solutions, especially in the context of reference point approaches.

Following these considerations, the development and application of appropriate methods has become a main aspect of MCDM research since the 1970s such that today there exists a plethora of methods and variants of methods. Altogether, there should be significantly more than one hundred methods and variants of methods (Hanne 2001). This abundance of methodological approaches in MCDM appears to be pleasing.

5.4 Neural networks and multicriteria decision support

During the past 20 years neural networks have been discussed in a number of publications in connection with applications in MCDM. Earlier surveys of this field can be found in Tigka and Zopounidis (1998) and Hanne (2001). The first explicit studies are due to Wang and Bender (1991), Wang and Malakooti (1992), Wang (1993a,b 1994a,b), and Malakooti and Zhou (1994). All these studies and most of the studies up to now are dealing with MADM problems for which a feedforward network is used for learning a multiattribute utility function by employing a modified backpropagation algorithm. In these analyses, different types of multiattribute utility functions (e.g., additive, quadratic, and polynomial-exponential functions), fuzzy utility functions and also the preference data based on a pair-wise comparison of alternatives are used.

Thus, the typical main idea of applying a neural network is as follows: first, the decision maker needs to articulate his or her preferences for a not too large set of alternatives. Then, parameters of the neural network are determined during a learning phase. The training data are the attribute values of the evaluated alternatives together with a preference indicator (i.e., the utility of the alternative). These data serve the neural network as a training set for learning a scalar evaluation function for the alternatives. In a working phase, the neural network can then easily be employed to automatically evaluate a larger set of feasible alternatives so that an optimal one can be chosen finally. Possibly, there is a another iteration with the 'manual' evaluation of alternatives, the training of the neural network and its subsequent application if the prior learning of the decision maker's preferences was not sufficiently successful.

In most applications, it is assumed that the input of the neural network are the criteria values of an alternative. Then, after processing these values through one or several layers of hidden neurons, one output neuron calculates the result of the neural network. It is thus assumed, that the neural network serves the purpose of evaluating each alternative independently by a single scalar value. This is the typical idea of a utility-based approach which assumes that the neural network calculates as output the utility (also denoted as value) of an alternative a, i.e.,

$$S_{out} = U(a) = U(z_1, \ldots, z_q). \tag{5.16}$$

In Equation (5.16) it is assumed that the neural network has a single output neuron *out* and that alternative a has the criteria values $z_1, \ldots, z_q$. For the utility function, usually additional properties are required such as monotonicity (i.e., increasing in the case of maximization criteria) in each of the components. In some of the papers, theoretical aspects of the representability of utility functions by neural networks are analyzed, see, e.g., Zhou and Malakooti (1990) and Malakooti and Zhou (1994).

Many other MCDM approaches are based on the same concept of evaluating alternatives using a scalarizing function which, in some cases, may not be in accordance with the assumptions of utility theory. Some examples of such methods are simple additive weighting, reference point approaches, goal programming, conjunctive levels, the TOPSIS method and so on (Hanne 2001). The specific benefits of using neural networks in the light of deviation of human behavior from standard utility concepts is pointed out by Leven and Levine (1996).

Note that the working of some well-known MCDM methods cannot be expressed by such a scalarizing function. Such methods are often not based on the axiom of independence of irrelevant alternatives but require instead the data of all alternatives to be processed simultaneously during the evaluation of the alternatives. Prominent representatives of such methods are the outranking methods, the analytic hierarchy process (AHP) and similar approaches (Hanne 2001). However, a neural network which works like one of those methods could also be realized, by providing all input data of a problem, i.e., $n \cdot q$ input neurons which process the data of the decision matrix simultaneously. An according proposal for using such a network can be found in Hanne (2001).

Some of the studies work with a given specified utility function which serves the calculation of reference values for the training phase. Usually, such studies lead to quite good results (or small training and test errors, respectively) which is not so surprising when small sizes of the training set are used. Working with real-life decision makers and their possibly incomplete, changing, or inconsistently formulated preferences might lead to worse results. Sun *et al*. (1996) propose to apply a feedforward network (with backpropagation) in an interactive context for MODM. In the iteration loop, some representatively spread alternatives are presented to the decision maker for evaluation. A proposal for using backpropagation feedforward networks for supporting multicriteria multi-person decisions based on utility functions is given by Wang and Archer (1994).

While the majority of studies intend to use the neural network for supporting a decision maker who deals with an MCDM problem, some research is rather directed to replace a traditional MCDM method or a part of such a method by the neural network, e.g., for computational reasons. In some other studies the neural network serves a forecasting purpose (but not for the prediction of preferences) and is used in combination with another multicriteria approach in some integrated decision support system. Other research is not

directed towards multicriteria decision analysis in a general sense but focuses on the design problem of neural networks which can be formulated in a multicriteria sense. The next section is intended to give a more detailed survey of various studies dealing with MCDM and neural networks.

5.4.1 Review of neural network applications to MCDM problems

A study by Zhou and Malakooti (1990) is based on the idea of using a feedforward neural network to learn the utility function of a decision maker for solving an MCDM problem. The neural network is constructed in a straightforward manner: For each criterion value there is one input neuron. The outputs are processed by a layer of hidden neurons and aggregated by a single output neuron which is expected to calculate the utility value of an alternative. It is suggested to use the neural network in an interactive way as follows: from a larger set of alternatives, a smaller number (e.g., $q + 1$ where q is the number of criteria) is presented to the decision maker in order to evaluate them by assigning a utility value. Using these data together with the alternatives' criteria, the neural network is trained. After that all alternatives are evaluated using the neural network and the $q + 1$ best alternatives are presented to the decision maker. If the decision maker agrees with their evaluation, the best alternatives can be selected. Otherwise, the decision maker evaluates the presented alternatives as before and the above process repeats with the neural network continuing its training using the new input data.

In Malakooti and Raman (2000) neural networks are applied to a machine setup problem for cutting metal. The problem here is not to select an alternative according to multiple criteria but to determine such criteria (process outputs) according to several parameters (process inputs). In the analyzed case, three objectives are relevant: the minimization of costs, the maximization of productivity, and the maximization of surface quality. Instead of working with these objectives, four process outputs are considered: the cutting force, the cutting temperature, the tool life, and the surface roughness. The process outputs depend on three parameters (process inputs): the cutting speed, the feed rate, and the depth of cut. Although, the relationship between input and output values can be expressed analytically, the authors suggest to let a feedforward neural network learn this relationship by using a backpropagation algorithm. It is also possible to apply a second neural network for learning the reverse relationship, i.e., to calculate input parameters when a most desired setting of output parameters has been specified by a decision maker. It is suggested first to present calculated process output based on a small number of different input settings (alternatives) to the decision maker, who selects a most preferred solution. In a second step, these alternatives are to be improved, using the second neural network for obtaining corresponding input values. The original objective functions are, however, not taken into account during this process. A similar approach based on approximating a multiattribute value function for determining cutting parameters is presented in Zuper and Cus (2003).

In Wang *et al*. (1994) the problem of prioritizing job shops is considered. Three criteria for determining the order acceptance sequence are taken into account: the profit rate, the slack time, and the customer credit. It is suggested to solve the problem using a feedforward neural network with one hidden layer. Unlike in standard topologies, the input layer neurons are not only connected with those of the hidden layer but also directly with the output neuron. The network employs six input neurons which are expected to

receive criteria values from two alternatives. The number of hidden neurons is six as well and the single output neuron is expected to calculate a comparison values for the two considered alternatives.

The neural network approach is tested using input value combinations (normalized criteria values) from 10 alternatives. Reference data for the learning process are obtained using an explicit multiattribute utility function. The neural network is then tested using further data from another 10 alternatives. The results lead to a small error measure and a priority list identical to that based on the multiattribute utility function.

In Stam *et al*. (1995, 1996) the usage of a neural network is suggested in connection with the AHP approach for MADM. It is assumed that a feedforward neural network works like the AHP: based on the pairwise comparison of the considered alternatives, the priority values for all alternatives are to be calculated. Assuming that there are n alternatives, $n(n-1)/2$ pairwise comparison values (neglecting the reversed comparisons and self-comparisons of alternatives) are used as input values. These data are processed by two layers of hidden neurons. n output neurons are then applied for calculating the evaluation of the alternatives.

Sun *et al*. (2000) suggest a new interactive approach for MCDM based on the utilization of a feedforward neural network. The problem under consideration is a MODM problem. The new method called FFANN-2 is an improvement of Sun *et al*. (1996) and assumes a scalarization of the multiple objective functions using a weighted Chebyshev (Tchebycheff) approach. In more detail, the approach works as follows: after some initialization and preliminary calculations, first a number of random weighting vectors (e.g., $20q$ where q is the number of objectives) is generated. Using these weights, the Chebyshev approach is employed to calculate nondominated solutions corresponding to each of the weights. A most widely dispersed subset of these solutions is presented to the decision maker who assigns scalar values to each of them, either directly or using some approach based on pairwise comparisons (i.e., in an AHP-like fashion). If the decision maker is not already satisfied with the best found solution, a neural network is used to learn the relationship between (rescaled) criterion values and the scalar evaluation of an alternative using the previously evaluated data. Then the considered set of weights is reduced and new random weights are generated. The Chebyshev approach is then used again to evaluate these solutions and the neural network is applied to evaluate them as well according to the learned preference function. This information is used to reduce the set of solutions which is then again presented to the decision maker for 'manual' evaluation. This process repeats again and again until the decision maker is satisfied with the best obtained solution or until a maximum number of iterations is conducted.

The employed feedforward network has a number of input neurons corresponding to the number of objectives. These input values are further processed by a layer of hidden neurons. One output neuron is assumed to calculate the overall evaluation corresponding to the input criteria values. The structure of the neural networks is not the standard topology but corresponds to Wang *et al*. (1994), where some computational experiments are described including a comparison with two other approaches which are mostly outperformed by the new methodology. The computation experiments are not done with a human decision maker but by an assumed utility function (four types are taken into consideration). The considered MODM problems are linear problems with a varying number of objectives, decision variables, and constraints. In the experiments the number of hidden neurons is varied between 0 and 6.

Another approach similar to that of Sun *et al*. (1996) is suggested in Huang *et al*. (2005). The proposed procedure can handle method parameters in a more general sense, i.e., not only weights in the context of the Chebyshev approach. Another difference is the utilization of the neural network: the preference function represented by the trained neural network is optimized. Also the concept of updating the weight space is somewhat different. The suggested approach requires a number of scalar optimization problems to be solved (e.g., optimization of an augmented Chebyshev distance); these are solved using a genetic algorithm.

Huang *et al*. (2005) suggest using that method for problems in the area of reliability optimization. Its suitability is demonstrated using an example problem with three objective functions, four variables, and a number of constraints. For the employed feedforward network three hidden neurons appear to be sufficient.

In Subbu *et al*. (2006, 2007) a decision support system for managing power plants based on multiple criteria is taken into account. The decision support system consists of three parts: a predictive model for forecasting future values of uncontrollable parameters, a performance function for determining objective values based on uncontrollable and controllable inputs, and a multiobjective optimizer for determining the most suitable (controllable) input values. The multiobjective optimizer uses an evolutionary algorithm for determining a set of approximated Pareto-optimal solutions to a given multiobjective optimization problem. This result is further evaluated using a decision function for determining a most preferred solution.

The neural network in this application is not directly connected with the MCDM part. It is used to represent the nonlinear mapping between (controllable and uncontrollable) input values and the objective functions. Two of the objective functions are to be minimized (NO_x emission and heat rate which is inversely related to efficiency) whereas two others (CO emission and load) are treated as constraints. Further constraints are used to introduce bounds for the variables' values. Based on the evolutionary algorithm a two-dimensional Pareto front of feasible solutions is then generated and visualized. Based on this result and using the decision function, solutions are recommended to the plant operator.

In Taha and Rostam (2011) the problem of selecting a machine (e.g., a CNC machine) for a flexible manufacturing cell is considered. It is suggested to solve this MADM using the fuzzy analytic hierarchy process (fuzzy AHP) to take care of uncertain or qualitative criteria. Due to these difficulties and because of possible biases in the decision making process, Taha and Rostam (2011) suggest verifying the results of the fuzzy AHP by using a neural network. A numerical example with four alternatives and 10 criteria is considered.

The feedforward neural networks consist of three layers: an input layer with 10 neurons (= number of criteria), a hidden layer with the number of neurons varied (7, 10, or 20), and an output layer with 4 neurons. Different variants of the neural network (with different settings for the transfer function and the number of learning iterations) are considered. A setting with several (here: 5) decision makers is considered as well. As input for the neural networks the weight for each criterion is used, as output the resulting priorities for the four alternatives for the considered decision maker are to be computed. Information about the specific MADM problem is implicitly stored in the network parameters. As a result the suggested neural network generates an output with a small error value.

The study by Kuo *et al*. (2002) is based on the idea of using a neural network in the context of a fuzzy AHP application. Here, the application problem is different, a convenience store location problem, but also the application scenario differs: the store locations are evaluated according to 43 criteria which are aggregated using the fuzzy AHP approach based on input from experts. The evaluation criteria from 34 existing stores are determined together with the number of visiting customers per day as an overall performance indicator. After some data standardization 30 data samples from those are then used as training data for a feedforward neural network. This network has 43 input values (corresponding to each of the criteria), a layer of hidden neurons, and one output neuron which is expected to predict the number of visiting customers per day. The data of the remaining 4 stores is then used as a testing sample. The training takes place using the common error backpropagation approach. The results of that approach are quite promising since there is a low prediction error, and the ranking of the 4 sample stores is identical to the actual data. The fuzzy AHP is then suggested to simplify the application of the approach with respect to data acquisition. It is tested to see how well the neural network approach works when only 31 or, respectively, 14 most important criteria according to the fuzzy AHP results are employed. The result is that the prediction error becomes worse but is still sufficiently good for practical application. The results of the neural network are also compared with those of a regression approach which leads to worse results

In Li-li and Yan (2007) a similar MCDM approach based on neural networks is used for solving a location problem for logistics centers. As a MADM method the fuzzy AHP and a feedforward neural network are employed as follows: first the criteria values are standardized to values between 0 and 1. A subset of alternatives is then evaluated using the fuzzy AHP. The results are used for the training data of the neural network. Moreover, the criteria weights obtained by the fuzzy AHP serve as initial weights for the neural network (for avoiding a local, nonglobal optimum and for a better interpretation of the weights). For evaluating further alternatives the neural network can then substitute the fuzzy AHP which is quite demanding for a decision maker.

A considered example location problem consisted of 20 alternatives and eight criteria which correspond to four high-level criteria. The hidden neurons employ a sigmoid function as an activation function whereas the output neuron used a linear activation function. Ten alternatives together with their obtained evaluations serve as training data for the neural network. The data of the remaining 10 alternatives serves as test data. The considered neural network consists of eight input neurons (corresponding to the eight criteria). The improvement of the connection weights is done using the backpropagation algorithm. The number of hidden neurons is determined in a trial and error fashion. An error value for the test data or the suitability of the obtained results for a decision maker is, however, not discussed in the paper.

Another facility location problem is considered in Chi *et al*. (1996). Using a modified AHP approach 16 location criteria are grouped into four categories (government policies, employment costs, labor costs, and productivity of resources) and priority weights are calculated. The basic MADM concept employs a reference point approach called the TOPSIS method. While the AHP is utilized to calculate weights for each of the criteria, the TOPSIS method is employed to calculate a scalar value for each of the alternatives based on these weights.

The suggested neural network is assumed to get the weighted criteria values of an alternative as input values. The neural network is, however, not expected to calculate a preference measure corresponding to that obtained by the TOPSIS method. Instead, it is rather used to provide a classification of alternatives. Five classification groups are considered in the paper. Instead of using a single output value which serves in the classification, five output values each of them corresponding to one of the groups appeared to be more useful. When applying the neural network, a considered alternative is assigned to the group where it has the highest output value in the corresponding neuron.

The reason for that concept lies in the application scenario for the method. It is assumed that location information is stored in a large database possibly with inconsistent or incomplete information. Therefore, the method is not assumed to make a definite selection of a best location but to reduce the set of interesting locations which can be further evaluated in detail by a decision maker. The suitability of the approach is demonstrated using some computational experiments.

In Bhattacharya *et al*. (2007) the problem of selecting a flexible manufacturing system is considered. The respective MADM problem consists of six alternatives which are evaluated according to six quantitative cost criteria and five qualitative criteria. The cost criteria are aggregated to a total cost criterion whereas the five qualitative criteria are aggregated using the AHP approach. The resulting two high-level criteria are standardized and express an objective factor measure and a subjective factor measure which are further aggregated to a single preference measure using a weighted average. A feedforward neural network approach is then used for learning the final preference measure. Here the learning is based on various variants of evolutionary algorithms which are, in particular, used for learning the network size (i.e., the number of hidden neurons). Backpropagation is used as a local search method for determining the weights of the neural network and, thus, for evaluating its fitness during the evolutionary computation. Compared with a traditional neural network (without evolutionary learning) and two other approaches, the suggested method performs quite well. Due to the small academic example problem and a missing involvement of real decision makers, the suitability for a practical application remains, however, unclear.

Chan *et al*. (2000) and Chan and Jiang (2001) present another example of applying the AHP and neural networks in an integrated approach. The considered application problem is the design of flexible manufacturing systems. Nineteen criteria are taken into account. Using the fuzzy AHP methodology these criteria can be aggregated into five high level criteria: finance, productivity, flexibility, customer satisfaction, and risk. Using the neural network approach from Stam *et al*. (1996) values from the values of the pairwise comparison of alternatives serve as an input. The neural network is then used for approximating the evaluations of the considered alternatives.

Wei and Jinfu (2008) suggest a fuzzy wavelet neural network to solve MADM problems. A wavelet neural network is a typical multilayer feedforward network with a discrete wavelet activation function which is similar to a sigmoid function but has two parameters, a scaling factor and a displacement factor. This approach is further ‘fuzzyfied’ by assuming triangular fuzzy numbers as input. For further processing, these fuzzy numbers are, however, transformed into crisp numbers. The AHP approach is used for generating training data for the neural network and for determining the initial weights of the network.

The approach is demonstrated using a small numerical example MADM problem with nine alternatives and seven criteria. The employed feedforward neural network then has seven input nodes (corresponding to the criteria), eight hidden nodes, and one output node. The obtained results for the training data are very similar to those obtained by the AHP (test data are not employed).

In Chen and Lin (2004) the application of neural networks for learning a multiattribute utility function is considered. The authors suggest a specific type of neural network called a decision neural network (DNN) which basically consists of two regular feedforward neural networks with identical structure, each of them with one output neuron. The two outputs of the networks are then combined by dividing one output by the other. The purpose of the two networks is as follows: each of them is assumed to calculate the utility of one alternative and the combined output then delivers a result corresponding to the pairwise comparison of the two alternatives. It is assumed that decision makers are usually better able to make such pairwise comparisons than to directly evaluate an alternative's utility.

For the application of the approach, it is suggested first to standardize the criteria evaluations of the alternatives. For a reasonable set of alternatives, the decision maker is then asked to make pairwise comparisons. These pairs of alternatives' criteria together with the comparison values are used for the training of the neural network. The training is done using the error backpropagation algorithm. After that the quality of network results is tested, and in the case of insufficient results, further alternatives are to be compared by the decision maker and used for a further training of the network.

Since Chen and Lin (2004) suggest using that approach for MODM problems the obtained neural network (only the feedforward subnetwork of the DNN) reflects the decision maker's utility function. Now any nonlinear (single-objective) optimization method can be used to calculate an optimum solution for the considered problem. The authors illustrate their approach with a numerical example and compare it with the feedforward artificial neural network by Sun *et al*. (1996). The results show that the DNN approach works with fewer comparisons and leads to slightly better values.

In Yazgan *et al*. (2009) a variant of the AHP, the analytic network process (ANP), is used to support the selection of ERP software. Using an ANP, the considered problem is structured. The problem consists of 17 criteria which are grouped into five categories. A small dummy problem with four alternatives (i.e., ERP software packages) is considered. A neural network with 17 input neurons, a layer of hidden neurons, and four output neurons is used for learning the evaluations of the considered four alternatives. As in a few other cases, this type of neural network cannot be utilized to evaluate a larger set of so far unconsidered alternatives but is specifically constructed and trained for the problem under consideration.

In Hanne (1994, 1997, 2001) it is discussed how neural networks can directly be constructed, i.e., without a previous learning process, so that they work like some well-established MCDM methods. For instance, neural networks with the functionality of the TOPSIS method, of the conjunctive levels method, or of a reference point approach are considered. In addition, neural networks with a given topology are considered and evolutionary learning processes are applied to design and adapt the network. As training data, either data coming from some MCDM approach or data resulting as ex-post quality indicator of alternatives are considered.

Apart from that Hanne (2001) includes concepts of using more general network structures than typical neural networks in decision processes. Nodes of such complex networks could correspond to single MCDM methods so that the network allows to combine results from different MCDM methods. This approach allows to deal with the so-called meta decision problem, i.e., which method should be used for a given decision problem or how to construct a best fitting method in such a situation. Another nonstandard neural network approach (adaptive resonance theory networks) in connection with MCDM is due to Leven and Levine (1996).

Shih *et al*. (2004) present a neural network approach to solve multilevel optimization problems which can be considered as a special case in multiobjective optimization. Using a Karush–Kuhn–Tucker transformation, the original mathematical problem is transformed into a single objective optimization problem. Then using a penalty function approach, this constrained problem is transformed into an unconstrained optimization problem. The neural network is then used to approximate the resulting nonlinear objective function. For regular multiobjective optimization problems, it is suggested to use a reference point approach for turning them into single objective optimization problems. The suitability of the suggested concept is demonstrated using three mathematical test problems. In the approach the purpose of the neural network is not to alleviate the burden of a decision maker but to find a good and standardized substitute formulation of an optimization problem (i.e., the neural network representation can be realized by a VLSI circuit).

In Abbass (2003) a completely different way of combining multiobjective optimization and neural networks is presented: it is assumed that for the neural network different functions could be optimized during the training phase, e.g., error functions based on different training samples, or traditional error functions and error functions with noise. This consideration turns the learning problem of the neural network into a multiobjective optimization problem. It is suggested to solve that problem by a method called Pareto differential evolution, which is a multiobjective variant of the differential evolution method. Thus, an evolutionary process of neural networks is simulated which is enriched by local search, i.e., new generated networks are improved by the backpropagation algorithm. The whole concept is called the Memetic Pareto Artificial Neural Network.

In García-Pedrajas *et al*. (2002) the performance evaluation of a neural network is considered in a multiobjective fashion. In the paper, a modular neural network type and the problem of determining its structure is considered. It is suggested to solve that problem by an evolutionary approach based on cooperation between the modules of the networks. Since it makes sense to consider the performance of each of the modules separately, a multiobjective overall performance measure is suggested. The network design problem is finally solved by an evolutionary multiobjective algorithm which is an adaption of the well-known Nondominated Sorting Genetic Algorithm approach in that field. The suitability of the suggested concept is demonstrated by some computational experiments.

5.4.2 Discussion

After a growing number of publications on using neural networks for MCDM which appeared during the 1990s, the interest in that area seemingly declined. There could be a number of reasons for this. On the one hand, we can observe a certain decline in neural network research in general. Today, approaches like support vector machines have become

new and powerful methods in the area of classification and, thus, major competitors of neural networks (see, e.g., Herbrich *et al*. 1999). On the other hand, neural networks can be considered as an established technology which requires less basic research than one or two decades ago. Instead, we find many successful applications in a real-life context.

Unfortunately, the latter only holds to a small degree for neural network applications in connection with MCDM. We have determined only two publications which deal with the combined application of neural networks and MCDM in a real-life industrial context. Therefore, the decline of interest should have additional causes. One of them might be the limited application scenario. As stated befores, usually the neural network is used to automate the evaluation of multicriteria alternatives according to the preferences of a decision maker. Therefore there must be a sufficiently large number of alternatives, or repeating decision problems, so that it makes sense to evaluate only a smaller subset of alternatives, employ the neural network for learning the preferences, and subsequently use it for evaluating the complete set of alternatives (or for the application to forthcoming similar decision problems). In many practical MADM applications, the number of alternatives, however, does not seem to be too large, so that this more complicated application scenario involving the neural network does not make much sense.

Another reason for the seemingly limited interest might be that often in publications related to neural networks, it might not become obvious to the researchers that they are dealing with an MCDM problem. They possibly just realize that they apply a neural network for a prediction or classification purpose without going into detail of what the nature of the predicted values might be (e.g., values resulting from physical, technical, psychological, or economic processes). In such cases it is difficult to recognize a related paper as a publication also related to MCDM. Some examples of this type of publication are Seo and Zhang (2000), Menchetti *et al*. (2003), Mota (2008), Aztiria *et al*. (2009) and Rigutini *et al*. (2011). In some of these publications the relationship to MCDM is quite clear although not mentioned. For instance, in Menchetti *et al*. (2003) the neural network is used to learn a multiattribute utility function. Other publications deal with the learning of human preferences in a different sense (e.g., based on search behavior) or deal with criteria which are not directly linked to criteria in the sense of MCDM.

5.5 Summary and conclusions

In this chapter we have outlined possibilities of applying neural networks in the area of MCDM. The main application purpose is to employ the neural network for learning a decision maker's utility function (or, in a more general sense, for learning a function which expresses the decision makers' preferences for alternatives which are characterized by several criteria). The main reason for that approach is to alleviate the decision maker's burden in evaluating all the alternatives of a decision problem by partially automating this process. Another reason might be to bring more consistency into that process by always applying the same evaluation function once it has been learned successfully by the neural network. (When a decision maker successively evaluates alternatives, his or her preferences might change or are just not sufficiently consistent.)

Although we have found a decreasing number of publications in this area during the last decade, the potential for utilizing such a combination is still there. Today, there is a growing desire for decision support systems and, in particular, systems which automate the corresponding processes, or at least, systems which efficiently guide a user through the

decision making process. For that reason, we expect that utilization of neural networks or related approaches (e.g., support vector machines) will be used more frequently in practical applications in the future. The basic research for such applications has been accomplished in various studies and along with the growing field of business intelligence and the increased business need for analytical methods, practitioners are becoming more aware of these approaches.

References

Abbass HA (2003) Pareto neuro-evolution: constructing ensemble of neural networks using multi-objective optimization. *The 2003 Congress on Evolutionary Computation*, vol. 3, pp. 2074–2080.

Aztiria A, Izaguirre A, Basagoiti R and Augusto J (2009) Learning about preferences and common behaviours of the user in an intelligent environment. In *Behaviour Monitoring and Interpretation – BMI–Smart Environments* (eds Gottfried B and Aghajan H), vol. 3. IOS Press, Amsterdam, pp. 289–315.

Baum EB and Haussler D (1989) What size net gives valid generalization? *Neural Computation* **1**, 151–160.

Bhattacharya A, Abraham A, Vasant P and Grosan C (2007) Evolutionary artificial neural network for selecting flexible manufacturing systems under disparate level-of-satisfaction of decision maker. *International Journal of Innovative Computing, Information and Control* **3**(1), 131–140.

Brause R (1991) *Neuronale Netze*. B. G. Teubner, Stuttgart.

Caporaletti LE, Dorsey RE, Johnson JD and Powell WA (1994) A decision support system for in-sample simultaneous equation systems forcasting using artificial neural networks. *Decision Support Systems* **11**, 481–495.

Chan FTS and Jiang B (2001) The applications of flexible manufacturing technologies in business process reengineering. *International Journal of Flexible Manufacturing Systems* **13**(2), 131–144.

Chan FTS, Jiang B and Tang NKH (2000) The development of intelligent decision support tools to aid the design of flexible manufacturing systems. *International Journal of Production Economics* **65**(1), 73–84.

Chen J and Lin S (2004) A neural network approach-decision neural network (DNN) for preference assessment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **34**(2), 219–225.

Chi SC, Benjamin CO and Riordan CA (1996) A neuro-computing group decision support system for identifying facility location. *IEEE International Conference on Systems, Man, and Cybernetics, 1996*, vol. 3, pp. 2059–2064.

Chu CH and Widjaja D (1994) Neural network system for forecasting method selection. *Decision Support Systems* **12**, 13–24.

Dutta S and Shekhar S (1988) Bond rating: A nonconservative application of neural networks. *IEEE International Conference on Neural Networks, 1988*, pp. 443–450.

Dutta S, Shekhar S and Wong MY (1994) Decision support in non-conservative domains: Generalizations with neural networks. *Decision Support Systems* **11**, 527–544.

Gal T (1995) *Postoptimal Analyses, Parametric Programming, and Related Topics. Degeneracy, Multicriteria Decision Making, Redundancy*, 2nd edn. De Gruyter, Berlin.

García-Pedrajas N, Hervás-Martínez C and Muñoz Pérez J (2002) Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks). *Neural Networks* **15**(10), 1259–1278.

Hanne T (1994) Die Integration von Multikriteria-Verfahren insbesondere mittels neuronaler Netze. *OR Spektrum* **16**, 277–283.

Hanne T (1997) Decision support for MCDM that is neural network-based and can learn. In *Multicriteria Analysis: Proceedings of the XIth International Conference on MCDM* (eds Climaco J). Springer, Berlin, pp. 401–410.

Hanne T (2001) *Intelligent Strategies for Meta Multiple Criteria Decision Making*. Springer, Berlin.

Hecht-Nielsen R (1987) Kolmogorov's mapping neural network existence theorem. In *Proceedings of IEEE First Annual International Conference on Neural Networks*, vol. 3, pp. 11–13.

Herbrich R, Keilbach M, Graepel T, Bollmann-Sdorra P and Obermayer K (1999) Neural networks in economics: Background, applications and new developments. In *Computational Techniques for Modelling Learning in Economics* (ed. Brenner T), pp. 169–196. Kluwer, Berlin.

Hill T and Remus W (1994) Neural network models for intelligent support of managerial decision making. *Decision Support Systems* **11**, 449–459.

Hornik K, Stinchcombe M and White H (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366.

Huang HZ, Tian Z and Zuo MB (2005) Intelligent interactive multiobjective optimization method and its application to reliability optimization. *IIE Transactions* **37**(11), 983–993.

Hwang CL and Masud ASM (1979) *Multiple Objective Decision Making – Methods and Applications*. Springer, Berlin.

Hwang CL and Yoon K (1981) *Multiple Attribute Decision Making - Methods and Applications*. Springer, Berlin.

Kamijo KI and Tanigawa T (1990) Stock price pattern recognition – a recurrent neural network approach. *International Joint Conference on Neural Networks*, vol. 1, pp. 215–221.

Kimoto T, Asakawa K, Yoda M and Takeoka M (1993) Stock market prediction system with modular neural networks. In *Neural Networks in Finance and Investing* (eds Trippi RR and Turban E). Probus Publishing, Chicago, pp. 343–357.

Kohonen T (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**, 59–69.

Kuo RJ, Chi SC and Kao S (2002) A decision support system for selecting convenience store location through integration of fuzzy AHP and artificial neural network. *Computers in Industry* **47**(2), 199–214.

Lapedes A and Farber R (1988) *How neural nets work Evolution, Learning and Cognition*. World Scientific, Singapore, pp. 331–346.

Leven SJ and Levine DS (1996) Multiattribute decision making in context: A dynamic neural network methodology. *Cognitive Science* **20**(2), 271–299.

Li-li Q and Yan C (2007) An interactive integrated MCDM based on FANN and application in the selection of logistic center location. *International Conference on Management Science and Engineering, ICMSE 2007*, pp. 162–167.

Malakooti B and Raman V (2000) An interactive multi-objective artificial neural network approach for machine setup optimization. *Journal of Intelligent Manufacturing* **11**, 41–50.

Malakooti B and Zhou YQ (1994) Feedforward artificial neural networks for solving discrete multiple criteria decision making problems. *Management Science* **40**(11), 1542–1561.

McCulloch WS and Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **7**, 115–133.

Menchetti S, Costa F, Frasconi P and Pontil M (2003) Comparing convolution kernels and recursive neural networks for learning preferences on structured data. In *Proceedings of IAPR – TC3 International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR 2003).*

Minsky M (1967) *Computation: Finite and Infinite Machines*. Prentice-Hall, Englewood Cliffs, NJ.

Mota J (2008) Using learning styles and neural networks as an approach to elearning content and layout adaptation. *DSIE'08 – Doctoral Symposium on Informatics Engineering*.

Müller B and Reinhardt J (1990) *Neural Networks: An Introduction*. Springer, Berlin.

Newell A (1981) Physical symbol systems. In *Perspectives on Cognitive Science* (ed. Norman DA). Ablex Publications, Norwood, MA, pp. 37–85.

Pau I and Jones RD (1994) A neural net model for prediction. *Journal of the American Statistical Association* **89**(424), 117–121.

Piramuthu S, Shaw MJ and Gentry JA (1994) A classification approach using multi-layered neural networks. *Decision Support Systems* **11**, 509–525.

Rigutini L, Papini T, Maggini M and Scarselli F (2011) SortNet: Learning to rank by a neural preference function. *IEEE Transactions on Neural Networks* **22**(9), 1368–1380.

Ritter H, Martinetz T and Schulten K (1991) *Neuronale Netze: Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*, 2nd edn. Addison Wesley, Bonn.

Rojas R (1993) *Theorie der neuronalen Netze: Eine systematische Einführung*. Springer, Berlin.

Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **65**, 386–408.

Rosenblatt F (1962) *Principles of Neuro-Dynamics*. Spartan, Washington, DC.

Schocken S and Ariav G (1994) Neural networks for decision support: Problems and opportunities. *Decision Support Systems* **11**, 393–414.

Schwefel HP (1981) *Numerical Optimization of Computer Models*. John Wiley & Sons, Ltd, Chichester.

Seo YW and Zhang BT (2000) Learning user's preferences by analyzing web-browsing behaviors. In *Proceedings of the 4th International Conference on Autonomous Agents*, *AGENTS '00*. ACM, New York, NY, pp. 381–387.

Sexton RS, Sriram RS and Etheridge H (2003) Improving decision effectiveness of artificial neural networks: A modified genetic algorithm approach. *Decision Sciences* **34**(3), 421–442.

Sharda R (1994) Neural networks for the MS/OR analyst: An application bibliography. *Interfaces* **24**(2), 116–130.

Shih HS, Wen UP, Lee S, Lan KM and Hsiao HC (2004) A neural network approach to multiobjective and multilevel programming problems. *Computers & Mathematics with Applications* **48**(1–2), 95–108.

Sohl JE and Venkatachalam AR (1995) A neural network approach to forecasting model selection. *Information & Management* **29**(6), 297–303.

Stam A, Sun M and Haines M (1995) Artificial neural network representations for hierarchical preference structures. Working paper WP-95-33, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Stam A, Sun M and Haines M (1996) Artificial neural network representations for hierarchical preference structures. *Computers & Operations Research* **23**(12), 1191–1201.

Steuer RE (1986) *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, Ltd, New York.

Subbu R, Bonissone P, Bollapragada S, Chalermkraivuth K, Eklund N, Iyer N, Shah R, Feng X and Weizhong Y (2007) A review of two industrial deployments of multi-criteria decision-making systems at general electric. *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, pp. 136–145.

Subbu R, Bonissone P, Eklund N, Weizhong Y, Iyer N, Feng X and Shah R (2006) Management of complex dynamic systems based on model-predictive multi-objective optimization *Proceedings of 2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pp. 64–69.

Sun M, Stam A and Steuer RE (1996) Solving multiple objective programming problems using feed-forward artificial neural networks: The interactive FFANN procedure. *Management Science* **42**(6), 835–849.

Sun M, Stam A and Steuer RE (2000) Interactive multiple objective programming using tchebycheff programs and artificial neural networks. *Computers & Operations Research* **27**(7-8), 601–620.

Taha, Z and Rostam S (2011) A fuzzy AHP-ANN-based decision support system for machine tool selection in a flexible manufacturing cell. *The International Journal of Advanced Manufacturing Technology* **57**(5–8), 719–733.

Tam KY (1994) Neural networks for decision support. *Decision Support Systems* **11**, 389–392.

Tigka KK and Zopounidis C (1998) Artificial neural networks systems for multiple criteria decision making. In *Managing in Uncertainty: Theory and Practice* (eds Zopounidis C and Pardalos PM). Kluwer Academic Publishers, Berlin, pp. 275–291.

Wang J (1993a) Multiple-objective optimisation of machining operations based on neural networks. *International Journal of Advanced Manufacturing Technology* **8**, 235–243.

Wang J (1993b) A neural network approach to multiple-objective cutting parameter optimization based on fuzzy preference information. *Computers and Industrial Engineering* **25**, 389–392.

Wang J (1994a) Artificial neural networks versus natural neural networks. a connectionist paradigm for preference assessment. *Decision Support Systems* **11**, 415–429.

Wang J (1994b) A neural network approach to modeling fuzzy preference relations for multiple criteria decision making. *Computers & Operations Research* **21**(9), 991–1000.

Wang J and Bender M (1991) Connectionist decision support systems for multiple criteria decision making. *1991 IEEE International Conference on Systems, Man, and Cybernetics, 1991. 'Decision Aiding for Complex Systems'*, vol. 3, pp. 1955–1960.

Wang J and Malakooti B (1992) A feedforward neural network for multiple criteria decision making. *Computers & Operations Research* **19**(2), 151–167.

Wang J, Yang JQ and Lee H (1994) Multicriteria order acceptance decision support in over-demanded job shops: A neural network approach. *Mathematical and Computer Modelling* **19**(5), 1–19.

Wang S and Archer NP (1994) A neural network technique in modeling multiple criteria multiple person decision making. *Computers & Operations Research* **21**(2), 127–142.

Wei Z and Jinfu Z (2008) Using fuzzy wavelet neural network to solve mcdm problem. *International Symposium on Computer Science and Computational Technology, ISCSCT '08*, vol. 2, pp. 458–461.

White H (1988) Economic prediction using neural networks: the case of IBM daily stock returns. *IEEE International Conference on Neural Networks*, vol. 2, pp. 451–458.

Wilson RL (1994) A neural network approach to decision alternative prioritization. *Decision Support Systems* **11**, 431–447.

Yazgan HR, Boran S and Goztepe K (2009) An ERP software selection process with using artificial neural network based on analytic network process approach. *Expert Systems with Applications* **36**(5), 9214–9222.

Yoon Y, Guimaraes T and Swales G (1994) Integrating artificial neural networks with rule-based expert systems. *Decision Support Systems* **11**, 497–507.

Yoon Y and Swales G (1993) Predicting stock price performance: A neural network approach. In *Neural Networks in Finance and Investing* (eds Trippi RR and Turban E). Probus Publishing, Chicago, pp. 329–341.

Zeleny M (1982) *Multiple Criteria Decision Making*. McGraw-Hill, New York.

Zhou Y and Malakooti B (1990) An adaptive feedforward artificial neural network with the applications to multiple criteria decision making. *IEEE International Conference on Systems, Man and Cybernetics*, pp. 164–169.

Zuper U and Cus F (2003) Optimization of cutting conditions during cutting by using neural networks. *Robotics and Computer-Integrated Manufacturing* **19**, 189–199.

6

Rule-based approach to multicriteria ranking

Marcin Szeląg[1], Salvatore Greco[2] and Roman Słowiński[1,3]

[1]*Institute of Computing Science, Poznań University of Technology, Poland*
[2]*Department of Economics and Business, University of Catania, Italy*
[3]*Systems Research Institute, Polish Academy of Sciences, Poland*

6.1 Introduction

In this chapter, we present a methodology for dealing with a *multicriteria ranking problem* using a preference model in the form of a set of decision rules induced from decision examples. The ranking consists in ordering a set of *objects* (also called alternatives, solutions, acts, actions, options, candidates, ...) from the best to the worst, while these objects are evaluated from multiple points of view considered relevant for the problem at hand and called *criteria* (also called attributes, features, variables, ...). Multicriteria ranking problems constitute one of three main categories of decision problems considered in the field of multiple criteria decision aid (MCDA) (also called multiple criteria decision making), which are ranking, choice and sorting (also called ordinal classification).

An important step in MCDA concerns selection and construction of criteria used for evaluation of objects. They are functions with ordinal or cardinal scales, built on elementary features of the objects. The aim is to set up a family of criteria which makes the pairwise comparison of all objects in the considered set meaningful. The criteria are equipped with monotonic preference scales which specify the preference orders in their value sets.

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

For a given finite set of objects A, and for a finite set of criteria $G = \{g_1, \ldots, g_n\}$ giving evaluations $g_i(a)$ to all $a \in A$, $i = 1, \ldots, n$, the only objective information that comes out from comparison of these objects on multiple criteria is a *dominance relation* D over set A. Given $a, b \in A$, object a dominates object b, which is denoted by aDb, if and only if $g_i(a) \succeq g_i(b)$ for each $i = 1, \ldots, n$, where $\succeq$ means 'is at least as good as'. The dominance relation D is a partial preorder, i.e., a reflexive and transitive binary relation defined over A on the basis of evaluations $g_i(\cdot)$, $i = 1, \ldots, n$.

Apart from trivial cases, dominance relation D is rather poor and leaves many objects incomparable – these are all nondominated objects in set A. In order to enrich the dominance relation and make the objects in A more comparable, one needs additional information about the value system of the decision maker (DM), called *preference information*. This information is provided by the DM, possibly assisted by an analyst, and it permits to build a more or less explicit model of the DM's preferences, called a *preference model*. The preference model relates the decision to evaluations of the objects on the considered criteria. In other words, the preference model aggregates multicriteria evaluations of objects. It is inducing a preference structure in set A. A proper exploitation of this structure leads then to a *final recommendation* in terms of ranking of objects from set A.

It follows from the above that the preference information and the preference model are two crucial components of MCDA. Below, with respect to these two components, we review some recent trends in MCDA.

As to the preference information, it depends on the adopted methodology: prices and interest rates for cost–benefit analysis, cost coefficients in objectives and technological coefficients in constraints for mathematical programming, a training set of decision examples for neural networks and machine learning, substitution rates for a value function of Multi-Attribute Utility Theory, pairwise comparisons of objects in terms of intensity of preference for the analytic hierarchy process, attribute weights and several thresholds for ELECTRE methods, and so on [see the state-of-the-art survey by Figueira *et al*. (2005a)].

Very often the preference information is not easily definable. For example, this is the case for the price of many immaterial goods and for the interest rates in cost–benefit analysis, or the case for the coefficients of objectives and constraints in mathematical programming models. Even if the required information is easily definable, like a training set of decision examples for neural networks, it is often processed in a way which is not clear for the DM, such that he/she cannot see what the exact relations are between the provided information and the final recommendation. Consequently, very often the decision aiding method is perceived by the DM as a *black box* whose result has to be accepted because the analyst's authority guarantees that the result is 'right'. In this context, the aspiration of the DM to find good reasons to make a decision is frustrated and increases the need for a more transparent methodology in which the relation between the original information and the final recommendation is clearly shown. Such a transparent methodology has been called a *glass box* (Greco *et al*. 2008a). Typically, it uses a training set of decision examples as the input preference information.

The decision examples may either by provided by the DM on a set of real or hypothetical objects, or may come from observation of DM's past decisions. Such an approach follows the paradigm of inductive learning used in artificial intelligence (Michalski, 1983), or robust ordinal regression becoming popular in operational research (Greco *et al*. 2010b). It is also concordant with the principle of posterior rationality postulated by March (1988) since it emphasizes the discovery of the DM's intentions as an

interpretation of actions rather than as a priori position. This paradigm has been used to construct various preference models from decision examples, e.g., the general additive utility functions (Figueira *et al*. 2009; Greco *et al*. 2008b), the outranking relations (Greco *et al*. 2011; Mousseau and Słowiński, 1998), the monotonic decision trees (Giove *et al*. 2002), and the set of '*if* ..., *then* ...' decision rules (Greco *et al*. 2005).

Of particular interest is the last model based on decision rules –it has been introduced to decision analysis by Greco, Matarazzo and Słowiński (Greco *et al*. 1999a, 2001a; Słowiński *et al*. 2005). A popular saying attributed to Slovic (1975) is that 'people make decisions and then search for rules that justify their choices'. The rules explain the preferential attitude of the DM and enable understanding of the reasons of his/her past decisions. The recognition of the rules by the DM (Langley and Simon, 1998) justifies their use for decision support. So, the preference model in the form of rules derived from decision examples fulfills both explanation and recommendation goals of decision aiding.

Preference information given in terms of decision examples is often inconsistent. This explains the interest in *rough set theory* (Pawlak, 1991). Rough set theory permits to *structure the data set* such that sets of objects (or sets of pairs of objects) are represented by pairs of ordinary sets called *lower* and *upper approximations*. The differences between upper and lower approximations are called boundary sets, and their cardinalities indicate to what degree the data set is inconsistent. Induction of decision rules from data structured in this way permits certain or possible rules to be obtained (Pawlak and Słowiński, 1994; Słowiński, 1993).

As the classical definition of rough sets is based on an indiscernibility relation in the set of objects, it cannot handle inconsistency encountered in decision examples involving multicriteria evaluations. To deal with this kind of inconsistency, Greco, Matarazzo and Słowiński generalized the classical rough set approach, so as to take into account preference orders and monotonic relationships between evaluations on criteria and assignment to decision classes. This generalization, called the dominance-based rough set approach (DRSA), has been adapted to a large variety of decision problems, including the multicriteria ranking problem (Fortemps *et al*. 2008; Greco *et al*. 1999a, 2001a, 2008a, 2010a; Słowiński *et al*. 2005, 2009). Moreover, DRSA has been extended to some probabilistic approaches (Błaszczyński *et al*. 2006, 2009; Greco *et al*. 2001c; Inuiguchi and Yoshioka, 2006), relaxing the classical definitions of lower approximations.

The usefulness of DRSA goes beyond the frame of MCDA. This is because the type of monotonic relationships handled by DRSA is also meaningful for problems where preferences are not considered but a kind of monotonicity relating ordered attribute values is meaningful for the analysis of data at hand, e.g., 'the larger the mass and the smaller the distance, the larger the gravity'. Note, moreover, that DRSA can be adapted to discover rules from any kind of input data, exhibiting monotonic relationships which are unknown a priori and hold in some parts of the evaluation space only. This requires a proper noninvasive transformation of the data, permitting representation of both positive and negative monotonic relationships that are to be discovered (Błaszczyński *et al*. 2012).

In this chapter, we present a methodology for preference learning from decision examples. It employs an adaptation of DRSA to the multicriteria ranking problem (Greco *et al*. 1995, 1997, 1999a, 2001a; Słowiński *et al*. 2005, 2009). In this adaptation, decision examples have the form of *pairwise comparisons* of some reference objects. These pairwise comparisons, presented in a so-called *pairwise comparison table* (PCT), specify if a weak preference relation (called an *outranking relation* S) holds for the considered

pairs of reference objects or not (called a *nonoutranking relation* S^c). Thus, decision rules induced from rough approximations of comprehensive preference relations S and S^c also involve pairs of objects. They constitute the preference model of the DM who gave the pairwise comparisons. Application of these rules on a set A of objects to be ranked yields a specific preference structure on A, represented by a directed graph. In this graph, nodes correspond to objects from A while arcs correspond to relations S and S^c among objects. In order to pass from the preference structure to the recommended ranking of objects, one has to apply an exploitation procedure called a *ranking method*. In this chapter, we consider some desirable properties of several ranking methods which are supposed to be useful. From among these methods, we choose the one that has the best properties.

The remainder of this chapter is organized as follows. In Section 6.2, we present the setting of the considered multicriteria ranking problem. In Section 6.3, we introduce the concept of the PCT. Section 6.4 concerns rough approximation of two comprehensive preference relations specified by a DM – outranking relation S and nonoutranking relation S^c. In Section 6.5, we discuss induction of decision rules from rough approximations of S and S^c, and application of these rules on a set of objects to be ranked. Section 6.6 concerns exploitation of the preference structure resulting from application of decision rules. In this section, we focus on some desirable properties of several ranking methods and select the method that has the best properties. In Section 6.7, we present an illustrative example that demonstrates usefulness of the proposed approach. The last section concludes the chapter.

6.2 Problem setting

We consider a multicriteria ranking problem in which objects belonging to a finite set A have to be ranked, either completely or partially. In the first case, one aims at obtaining a *weak order* (also called a *complete preorder* or *total preorder*) over A, i.e., a binary relation which is reflexive, transitive, and complete. In the latter case, one accepts a *partial preorder* over A, i.e., a binary relation which is reflexive, and transitive, but noncomplete, in general. The objects from set A are evaluated by set $G = \{g_1, \ldots, g_n\}$ of n criteria. We assume that this set is a *consistent family of criteria* (Roy and Bouyssou, 1993), i.e., we assume that G satisfies the properties of *completeness* (all relevant criteria are considered), *monotonicity* (the better the evaluation of an object on considered criteria, the more it is preferable to another object), and *nonredundancy* (there is no criterion which could be removed without violating one of the previous two properties). Each *criterion* $g_i \in G, i = 1, \ldots, n$, is a real-valued function $g_i : A \to \Re$, with *cardinal* (i.e., interval or ratio) *scale* or *ordinal scale* (which is given a priori or results from an order-preserving number-coding of non-numerical ordinal evaluations). Thus, value $g_i(a)$, $a \in A$, represents the evaluation of object a with respect to (w.r.t.) criterion g_i. A criterion with the cardinal scale is called a *cardinal criterion*; the set of all cardinal criteria is denoted by $G^N \subseteq G$. A criterion with the ordinal scale is called an *ordinal criterion*; the set of all ordinal criteria is denoted by $G^O \subseteq G$. Moreover, $G^N \cup G^O = G$ and $G^N \cap G^O = \emptyset$. The meaning of the two scales is such that in the case of a criterion $g_i \in G^N$ with a cardinal scale, one can define a function $k_i : \Re^2 \to \Re$ which measures the *intensity of preference* (positive or negative) of object a over object b, taking into account evaluations $g_i(a), g_i(b),\ a, b \in A$. For properties of function k_i, and different ways of defining it, see Greco *et al*. (2001a). Basically, k_i is nondecreasing w.r.t. the first

argument, and nonincreasing w.r.t. the second argument. For the sake of simplicity, we assume in this chapter that for each cardinal criterion $g_i \in G^N$, intensity of preference is defined as: $k_i[g_i(a), g_i(b)] = \Delta_i(a, b) = g_i(a) - g_i(b)$. In the case of a criterion $g_i \in G^O$ with an ordinal scale, this is not possible (as differences of evaluations are not meaningful) and one can only establish an order of evaluations $g_i(a)$, $a \in A$.

We assume, moreover, without loss of generality, that all the criteria are of *gain type*, i.e., the greater the criterion value the better.

Let us denote by $V_{g_i} = \Re$ the value set (domain) of criterion $g_i \in G$. Then, set $V_G = \prod_{i=1,\ldots,n} V_{g_i}$ is called the *G-evaluation space*.

Given the statement of a multicriteria ranking problem, the only objective information one can get is the *dominance relation* D over the set of objects A, defined in the G-evaluation space. Let us consider objects $a, b \in A$; object a is said to dominate object b, denoted by aDb, if and only if (iff) for all $g_i \in G : g_i(a) \geq g_i(b)$. The dominance relation D is, however, too poor because it leaves many objects incomparable. In order to make the objects more comparable, a DM must supply some *preference information* revealing his/her value system w.r.t. multicriteria evaluations. We consider a frequent case, when the preference information has the form of *pairwise comparisons* of some objects relatively well known to the DM, called *reference objects*. This information is thus composed of some decision examples on the reference objects.

Let us denote by A^R the set of all reference objects. Set A^R can be a subset of A, however, it is not required by the presented methodology. If $A^R \not\subseteq A$, then we just need to define each criterion $g_i \in G, i = 1, \ldots, n$, as function $A \cup A^R \rightarrow \Re$, and dominance relation D over set $A \cup A^R$. In any case, $A \backslash A^R$ is a set of objects unseen during preference model learning. Following Greco *et al.* (1999a, 2001a) and Słowiński *et al*. (2005), we consider that for each ordered pair (a, b) of reference objects such that not aDb, the DM can state either 'object a is at least as good as object b' (in other words 'object a outranks object b'), or 'object a is not at least as good as object b' (in other words 'object a does not outrank object b'), or abstain from any judgment. The first situation is denoted by aSb [or $(a, b) \in S$, or $S(a, b) = 1$], while the second one is denoted by aS^Cb [or $(a, b) \in S^C$, or $S^C(a, b) = 1$]. Moreover, we fix aSb for pairs $(a, b) \in A^R \times A^R$ such that aDb. Thus, from a formal point of view, the DM can reveal his/her preferences by assigning pairs of objects to any of the two considered crisp *comprehensive preference relations* over set A^R: *outranking* relation S or *nonoutranking* relation S^C. Obviously, relation S is a weak preference relation which, in general, is only reflexive. It is not symmetric, not transitive, and not complete. Moreover, nonoutranking relation S^C is irreflexive, and, in general, it is not symmetric, not transitive, and not complete. This is to say that the preference information coming from the DM is relatively weak and nonexhaustive.

By expressing his/her preferences in the way described above, for each pair of objects $(a, b) \in A^R \times A^R$ the DM can easily specify any of the four situations typically considered in MCDA, i.e.:

- *strict preference* P:

$$aPb \Leftrightarrow aSb \wedge bS^Ca, \tag{6.1}$$

- *inverse strict preference* P^{-1}:

$$aP^{-1}b \Leftrightarrow aS^Cb \wedge bSa, \tag{6.2}$$

- *indifference* I:

$$aIb \Leftrightarrow aSb \wedge bSa, \tag{6.3}$$

- *incomparability* J:

$$aJb \Leftrightarrow aS^C b \wedge bS^C a. \tag{6.4}$$

Alternative elicitation of preference information in terms of pairwise comparisons employing graded comprehensive preference relations can be found in Fortemps *et al*. (2008).

It is worth stressing that expressing decision examples on the reference objects is cognitively relatively easy for the DM. In our approach, instead of requiring that the DM provides values of some difficult parameters like weights of criteria or different thresholds [see, e.g., methods from the ELECTRE family (Roy, 1991)], and then using this information in a complex preference model, we treat the decision examples supplied by the DM as the input data, and then follow with learning of a preference model of the DM in easy rule terms.

To simplify the notation, in the following, we will use unique symbol T to refer to any of the comprehensive preference relations S and S^C when these relations are considered jointly, unless this may cause misunderstanding. Moreover, we denote by $\mathcal{I}_G, \mathcal{I}_{G^N}, \mathcal{I}_{G^O} \subseteq \{1, \ldots, n\}$ the sets of indexes of criteria belonging to G, G^N, G^O, respectively, where $\mathcal{I}_{G^N} \cap \mathcal{I}_{G^O} = \emptyset$ and $\mathcal{I}_{G^N} \cup \mathcal{I}_{G^O} = \mathcal{I}_G$.

6.3 Pairwise comparison table

The preference information of the DM in the form of pairwise comparisons of reference objects is used to create a PCT, first introduced in Greco *et al*. (1995, 1997). Let us denote by $B \subseteq A^R \times A^R$ the set of pairs of reference objects for which the comprehensive preference of the DM is known. This set is composed of pairs $(a, b) \in A^R \times A^R$, such that not aDb, for which the DM expressed his/her preferences by declaring aSb or $aS^C b$, as well as of other pairs $(c, d) \in A^R \times A^R$, which are assigned to S because cDd.

Intuitively, a PCT created on the basis of preference information supplied by the DM is an $m \times (n + 1)$ data table, denoted by S_{PCT}, where m is the cardinality of set B of pairs. First n columns of this table correspond to criteria from set G. The last, $(n + 1)$th, column represents the comprehensive preference relation S or S^C. Each row of S_{PCT} corresponds to a pair of reference objects from B. As mentioned in Section 6.2, when comparing two objects $a, b \in A^R$ on a cardinal criterion $g_i \in G^N$, $i \in \mathcal{I}_{G^N}$, one puts in the corresponding column of S_{PCT} the difference $g_i(a) - g_i(b)$. When comparing two objects $a, b \in A^R$ on an ordinal criterion $g_i \in G^O$, $i \in \mathcal{I}_{G^O}$, one puts in the corresponding column of S_{PCT} an ordered pair of ordinal evaluations $(g_i(a), g_i(b))$.

Describing the PCT more formally, one can say that each pair of objects $(a, b) \in B$ is evaluated on set G of criteria, such that:

- for criterion $g_i \in G^N$, the evaluation of $(a, b) \in B$ is defined as $q_i(a, b) = g_i(a) - g_i(b) \in V_{q_i} = \Re$,
- for criterion $g_i \in G^O$, the evaluation of $(a, b) \in B$ is defined as $q_i(a, b) = (g_i(a), g_i(b)) \in V_{q_i} = \Re \times \Re$.

Then, set $V_Q = \prod_{i \in \mathcal{I}_G} V_{q_i}$ is called the *Q-evaluation space*.

6.4 Rough approximation of outranking and nonoutranking relations

In Section 6.2, we considered the dominance relation D over set of objects A, defined in the G-evaluation space. Here, we introduce another type of dominance relation, denoted by D_2. This binary relation is defined over set B of pairs of objects, in the Q-evaluation space. However, as it is more convenient, below we introduce relation D_2 using only the evaluations of objects from set A^R on the criteria from set G.

First, let us consider a case when set G is composed of cardinal criteria only, i.e., $G^N = G, G^O = \emptyset$. Then, given pairs of objects $(a, b), (c, d) \in B$, pair (a, b) is said to dominate pair (c, d) w.r.t. criteria from G [denoted by $(a, b)D_2(c, d)$] iff $\Delta_i(a, b) \geq \Delta_i(c, d)$ for each $g_i \in G$, where $\Delta_i(a, b)$ denotes $g_i(a) - g_i(b)$. Let D_2^i be the dominance relation over B confined to single criterion $g_i \in G$. This relation is reflexive, transitive and complete. Therefore, D_2^i is a weak order over B. Since an intersection of weak orders is a partial preorder, and relation D_2 over B is the intersection of relations D_2^i, $i \in \mathcal{I}_G$, then relation D_2 is a partial preorder over B.

Secondly, let us consider a case when set G is composed of ordinal criteria only, i.e., $G^O = G, G^N = \emptyset$. Then, given pairs of objects $(a, b), (c, d) \in B$, pair (a, b) is said to dominate pair (c, d) w.r.t. criteria from G iff $g_i(a) \geq g_i(c)$ and $g_i(b) \leq g_i(d)$ for each $g_i \in G$. In other words, pair (a, b) is said to dominate pair (c, d) w.r.t. criteria from G iff aDc and dDb. Let D_2^i be the dominance relation over B confined to single criterion $g_i \in G$. This relation is reflexive, transitive but noncomplete (i.e., it is possible that not $(a, b)D_2^i(c, d)$ and not $(c, d)D_2^i(a, b)$ for some $(a, b), (c, d) \in B$ and $g_i \in G$). Therefore, D_2^i is a partial preorder over B. Since an intersection of partial preorders is also a partial preorder, and relation D_2 over B is the intersection of relations D_2^i, $i \in \mathcal{I}_G$, then D_2 is a partial preorder over B.

Finally, when set G is composed of both cardinal and ordinal criteria, i.e., when $G^N \neq \emptyset$ and $G^O \neq \emptyset$, then given pairs of objects $(a, b), (c, d) \in B$, pair (a, b) is said to dominate pair (c, d) w.r.t. criteria from G iff (a, b) dominates (c, d) w.r.t. both G^N and G^O. Since the dominance w.r.t. G^N is a partial preorder over B and the dominance w.r.t. G^O is a partial preorder over B, then the dominance D_2, being the intersection of these two dominance relations, is also a partial preorder over B.

Let $G' \subseteq G$ and pairs $(a, b), (c, d) \in B$. Then, if (a, b) dominates (c, d) w.r.t. set G of criteria, then (a, b) dominates (c, d) w.r.t. set G'.

Given a pair of objects $(a, b) \in B$ we define the following:

- a set of pairs of objects dominating (a, b), called the *dominating set* or *positive dominance cone* in the Q-evaluation space:

$$D_2^+(a, b) = \{(c, d) \in B : (c, d)D_2(a, b)\}, \tag{6.5}$$

- a set of pairs of objects dominated by (a, b), called the *dominated set* or *negative dominance cone* in the Q-evaluation space:

$$D_2^-(a, b) = \{(c, d) \in B : (a, b)D_2(c, d)\}. \tag{6.6}$$

In Equation (6.5) and Equation (6.6), the pair of objects (a, b) is called an *origin* of the dominance cone. Positive and negative dominance cones are 'granules of knowledge' used to approximate outranking and nonoutranking relations, respectively.

We formulate the following *dominance principle* w.r.t. pairwise comparisons of the DM: 'if a is preferred to b at least as much as c is preferred to d with respect to each $g_i \in G$, then the comprehensive preference of a over b should not be weaker that the comprehensive preference of c over d'. This means that if $(a, b)D_2(c, d)$, one expects that:

(i) if $aS^C b$, then $cS^C d$;

(ii) if cSd, then aSb.

Violation of this dominance principle is considered as an *inconsistency* w.r.t. dominance relation D_2 over B.

In practice, decision examples given by a DM (who is a single person or a collectivity) are often inconsistent due to hesitation of the DM, unstable character of preferences, or some hidden additional aspects which one cannot express explicitly (e.g., Roy 1996). These inconsistencies cannot be considered as a simple error or as noise. They can convey important information that should be taken into account when constructing a preference model of the DM. Rather than correct or ignore these inconsistencies, we propose to handle them using the dominance-based rough set approach. Before learning of a preference model of the DM, we structure pairs of objects contained in S_{PCT} by calculation of lower approximations of comprehensive preference relations. In this way, we restrict a priori the set of pairs of objects on which the preference model is built to a subset of sufficiently consistent pairs of objects belonging to lower approximations. This restriction is motivated by a postulate for learning from (sufficiently) consistent data, so that the knowledge gained from this learning is relatively certain (or, in other words, the induced preference model is reliable). Analogous restriction proved to be effective in the case of ordinal classification problems with monotonicity constraints (Błaszczyński *et al*. 2010, 2011). It is worth underlining that, although only sufficiently consistent pairs of objects from S_{PCT} are used to construct a preference model of the DM, the remaining pairs of objects are not removed from S_{PCT}. In other words, the approach proposed in this chapter does not boil down to a simple pre-processing performed to remove inconsistent decision examples. In fact, inconsistent pairs of objects play the role of 'counterexamples', helping this way to induce a preference model.

In previous applications of DRSA to multicriteria ranking (Greco *et al*. 1999a, 2001a; Słowiński *et al*. 2005), outranking and nonoutranking relations were approximated using a strict inclusion relation between dominance cones originating in pairs of objects $(a, b) \in B$ and the comprehensive preference relations. Precisely, lower approximations of relations S and S^C were defined as:

$$\underline{S} = \{(a, b) \in B : D_2^+(a, b) \subseteq S\}, \tag{6.7}$$

$$\underline{S^C} = \{(a, b) \in B : D_2^-(a, b) \subseteq S^C\}. \tag{6.8}$$

These definitions of lower approximations appear to be too restrictive in practical applications. Consequently lower approximations of S and S^C are often empty, preventing generalization of pairwise comparisons in terms of sufficiently certain decision rules. Therefore, in this chapter, we apply variable consistency dominance-based rough set

approach (VC-DRSA) (Błaszczyński *et al*. 2009; Greco *et al*. 2001c), which is a probabilistic extension of the classical DRSA. Since originally VC-DRSA was introduced for ordinal classification problems, here we adapt its definitions of variable consistency lower approximations to the case of approximating outranking and nonoutranking relations. In the adapted definitions of variable consistency lower approximations of S and S^C, we use consistency measure $\epsilon_T : B \rightarrow [0, 1]$, introduced in Błaszczyński *et al*. (2007, 2009), defined as:

$$\epsilon_S(a, b) = \frac{|D_2^+(a, b) \cap S^C|}{|S^C|}, \tag{6.9}$$

$$\epsilon_{S^C}(a, b) = \frac{|D_2^-(a, b) \cap S|}{|S|}. \tag{6.10}$$

Given pair of objects $(a, b) \in B$ and relation T, value $\epsilon_T(a, b)$ reflects consistency of pair (a, b) w.r.t. T. ϵ_T is a cost-type consistency measure, which means that value zero denotes full consistency and the greater the value, the less consistent is a given pair of objects. The definitions of variable consistency lower approximations adapted to the case of approximating outranking and nonoutranking relations are the following:

$$\underline{S} = \{(a, b) \in S : \epsilon_S(a, b) \leq \theta_S\}, \tag{6.11}$$

$$\underline{S^C} = \{(a, b) \in S^C : \epsilon_{S^C}(a, b) \leq \theta_{S^C}\}, \tag{6.12}$$

where consistency thresholds $\theta_S, \theta_{S^C} \in [0, 1)$. It is worth noting that in case $\theta_S = \theta_{S^C} = 0$, the variable consistency lower approximations (6.11) and (6.12) are equal to the lower approximations (6.7) and (6.8), respectively. In the following, unless this may cause misunderstanding, we drop 'variable consistency' and call sets of pairs of objects defined by (6.11) and (6.12) just lower approximations of relations S and S^C, respectively.

In Błaszczyński *et al*. (2009) we defined several consistency measures. The choice of particular consistency measure ϵ_T is dictated by several factors. The first one is that measure ϵ_T features an easy interpretation – it can be interpreted as an estimate of conditional probability that a pair of objects $(c, d) \in B$ belongs to the dominance cone originating in pair $(a, b) \in B$ given that pair (c, d) does not belong to the considered comprehensive preference relation. The second factor is a good performance of this measure in computational experiments (Błaszczyński *et al*. 2010, 2011), comparing with other consistency measures. The third factor is the fact that measure ϵ_T has all monotonicity properties (Błaszczyński *et al*., 2009) relevant to the case of a PCT with just two possible decisions for each pair of objects, i.e., assignment to relation S or to S^C. Precisely, measure ϵ_T has properties: (m1) – monotonicity w.r.t. the set of criteria; (m2) – monotonicity w.r.t. relation T when T is augmented by new pairs of objects; and (m4) – monotonicity w.r.t. dominance relation D_2 over B.

Using definitions (6.11) and (6.12), one can define variable consistency upper approximations and variable consistency boundaries of sets S and S^C as in Błaszczyński *et al*., (2009).

We define *positive regions* of relations S and S^C as follows:

$$POS(S) = \bigcup_{(a,b) \in \underline{S}} D_2^+(a, b), \tag{6.13}$$

$$POS(S^C) = \bigcup_{(a,b)\in \underline{S^C}} D_2^-(a,b). \tag{6.14}$$

Positive regions defined above contain pairs of objects sufficiently consistent, i.e., belonging to lower approximations of relation S (6.11) or S^C (6.12), and can also contain some inconsistent pairs of objects which fall into dominance cones $D_2^+(\cdot,\cdot)$ or $D_2^-(\cdot,\cdot)$ originating in pairs of objects from lower approximations of relation S or S^C, respectively. Moreover, one can define boundary and negative regions of relations S and S^C analogously to Błaszczyński *et al*. (2006, 2011). It is also possible to perform further DRSA analysis by calculating the quality of approximation, reducts, and the core (Greco *et al*. 1999b, 2001a, 2005; Słowiński *et al*. 2005, 2009).

6.5 Induction and application of decision rules

After structuring decision examples supplied by the DM into lower approximations of comprehensive preference relations, we induce a generalized description of sufficiently consistent pairs of objects from S_{PCT} in terms of a set of *minimal decision rules*. An induced set of rules is considered to be a *preference model* of the DM who gave the pairwise comparisons of reference objects. Each rule is a statement of the type:

$$\textit{if}\ \ \Phi, \textit{then}\ \ \Psi,$$

where Φ and Ψ denote the *condition* and *decision* part of the rule, called also *premise* and *conclusion*, respectively. The condition part of the rule is a conjunction of elementary conditions concerning individual criteria, and the decision part of the rule suggests assignment of pairs of objects covered by the rule to outranking relation S or to nonoutranking relation S^C. The rule is said to *cover* a pair of objects $(a,b) \in A \times A$ if this pair satisfies all the elementary conditions of the rule. A pair of objects $(a,b) \in B$ is said to *support* the rule if this pair satisfies all the elementary conditions and the conclusion of the rule. Rule r_T, suggesting assignment of covered pairs of objects to relation T, is called *minimal* if there is no other rule r'_T having premise at least as general as that of r_T (i.e., employing a subset of elementary conditions of r_T and/or more general elementary conditions than r_T) and consistency not worse than that of r_T (where by consistency of rule r_T we understand the value of a rule consistency measure, as explained later in this section). In the following, a minimal decision rule is denoted in short as an *m-rule*. The interest in m-rules comes, obviously, from the fact that they generalize decision examples better than nonminimal rules. Thus, generation of m-rules may be seen as a way to avoid overfitting.

Decision rules are induced so as to cover pairs of objects from lower approximations (6.11) and (6.12). However, in some cases it is impossible for a rule to cover only pairs of objects from a lower approximation. To handle these cases, the positive region of the considered comprehensive preference relation is computed according to (6.13) or (6.14).

Set $\underline{T}$ of pairs of objects belonging to the lower approximation of comprehensive preference relation T is the basis for induction of a set of m-rules that suggest assignment to T. A rule from this set is supported by at least one pair of objects from $\underline{T}$, and it covers pair(s) of objects from $POS(T)$. The elementary conditions (selectors) that form this rule are built using only evaluations of objects present in the pairs of objects that belong to $\underline{T}$.

Below, we define the syntax of decision rules that generalize the description of sufficiently consistent pairs of objects present in a PCT:

$$\begin{gathered} \textit{if}\,(\Delta_{i_1}(a,b) \geq \delta_{i_1}) \wedge \ldots \wedge (\Delta_{i_p}(a,b) \geq \delta_{i_p}) \wedge \\ (g_{i_{p+1}}(a) \geq r_{i_{p+1}} \wedge g_{i_{p+1}}(b) \leq s_{i_{p+1}}) \wedge \ldots \wedge (g_{i_z}(a) \geq r_{i_z} \wedge g_{i_z}(b) \leq s_{i_z}), \\ \textit{then}\; aSb, \end{gathered} \tag{6.15}$$

$$\begin{gathered} \textit{if}\,(\Delta_{i_1}(a,b) \leq \delta_{i_1}) \wedge \ldots \wedge (\Delta_{i_p}(a,b) \leq \delta_{i_p}) \wedge \\ (g_{i_{p+1}}(a) \leq r_{i_{p+1}} \wedge g_{i_{p+1}}(b) \geq s_{i_{p+1}}) \wedge \ldots \wedge (g_{i_z}(a) \leq r_{i_z} \wedge g_{i_z}(b) \geq s_{i_z}), \\ \textit{then}\; aS^C b, \end{gathered} \tag{6.16}$$

where $\Delta_{i_j}(a,b)$ denotes $g_{i_j}(a) - g_{i_j}(b)$, $\delta_{i_j} \in \{g_{i_j}(c) - g_{i_j}(d) : (c,d) \in B\} \subseteq \Re$, for $i_j \in \{i_1, \ldots, i_p\} \subseteq \mathcal{I}_{G^N}$; $(r_{i_j}, s_{i_j}) \in \{(g_{i_j}(c), g_{i_j}(d)) : (c,d) \in B\} \subseteq \Re \times \Re$, for $i_j \in \{i_{p+1}, \ldots, i_z\} \subseteq \mathcal{I}_{G^O}$. For instance, considering ranking of cars, a decision rule could be 'if car *a* has maximum speed at least 25 km/h greater than car *b* (cardinal criterion) and car *a* has comfort at least 3 while car *b* has comfort at most 2 (ordinal criterion), then car *a* is at least as good as car *b*', where values 2 and 3 code ordinal evaluations medium and good, respectively.

The rules with syntax (6.15) are called *at least rules*, while the rules with syntax (6.16) are called *at most rules*. Let us observe that the above syntax of decision rules is concordant with the definition of dominance relation D_2 over B in the sense that the premise of each decision rule is a positive or negative dominance cone in the Q-evaluation space. Moreover, as we work with variable consistency lower approximations, in order to cover by rules all pairs of objects from $\underline{S}$ and $\underline{S^C}$, we have to agree that not all the rules will be fully consistent. For example, it is inevitable that a rule suggesting assignment to relation S covers pairs of objects that do not belong to S but dominate in the Q-evaluation space at least one pair of objects from $\underline{S}$ that supports the considered rule. Therefore, in the following, we speak about *probabilistic decision rules* to underline the fact that not all pairs of objects from S_{PCT} that are covered by a rule have to support this rule.

Decision rules can be characterized by many attractiveness measures [see Greco *et al.* (2004) for a study of some properties of these measures].

Since we work with probabilistic decision rules, it is important to control consistency of these rules. To this end, we define a cost-type *rule consistency measure* (Błaszczyński *et al.* 2010, 2011) denoted by $\widehat{\epsilon}_T$. This measure is a function $\widehat{\epsilon}_T : R_T \rightarrow [0,1]$, where R_T is the set of rules suggesting assignment to relation T. Let us denote by $\Phi(r_T)$, $\Psi(r_T)$, and $\|\Phi(r_T)\|$, the condition part of rule r_T, its decision part, and the set of pairs of objects covered by the rule, respectively. Then, measure $\widehat{\epsilon}_T$ is defined as:

$$\widehat{\epsilon}_T(r_T) = \frac{|\,\|\Phi(r_T)\| \cap \neg T\,|}{|\neg T|}, \tag{6.17}$$

where $\neg T = B \backslash T$ is the complement of relation T w.r.t. set B (obviously, $\neg S = S^C$ and $\neg S^C = S$).

Induced rules have to satisfy the same constraints on consistency as pairs of objects from the lower approximation which serves as a base for rule induction. In particular,

each rule r_T is required to satisfy threshold θ_T, i.e., $\widehat{\epsilon}_T(r_T)$ has to be not greater than θ_T. In the following, rule r_T satisfying threshold θ_T is called *sufficiently consistent* and denoted in short as the *sc-rule*. Since rule consistency measure $\widehat{\epsilon}_T$ is a counterpart of consistency measure ϵ_T defined as (6.9) and (6.10), it can be shown that $\widehat{\epsilon}_T$ derives monotonicity properties from ϵ_T.

Let us now remember some useful definitions concerning probabilistic decision rules, introduced in Błaszczyński *et al*. (2011).

A probabilistic decision rule r_T suggesting assignment to relation T is *discriminant* if it covers only pairs of objects belonging to positive region $POS(T)$. In the following, a discriminant decision rule is denoted in short as a *d-rule*. Moreover, rule r_T is *robust* if there exists a pair of objects $(a, b) \in \underline{T}$ which is a *base* of r_T. Considering for example definition (6.15), it means that $q_{i_1}(a, b) = \delta_{i_1} \wedge \ldots \wedge q_{i_p}(a, b) = \delta_{i_p} \wedge q_{i_{p+1}}(a, b) = (r_{i_{p+1}}, s_{i_{p+1}}) \wedge \ldots \wedge q_{i_z}(a, b) = (r_{i_z}, s_{i_z})$. In the following, a robust decision rule is denoted in short as an *r-rule*. Set R_T of rules suggesting assignment to relation T is *complete* iff each pair of objects $(a, b) \in \underline{T}$ is covered by at least one rule $r_T \in R_T$. In the following, a complete set of decision rules is denoted in short as a *c-set of rules*. Finally, set R_T is *minimal* if for each $r_T \in R_T$ set $R_T \backslash \{r_T\}$ is not complete. In the following, a minimal set of decision rules is denoted in short as an *m-set of rules*.

Induction of decision rules is a complex problem and many algorithms have been introduced to solve it. Examples of rule induction algorithms that were defined for DRSA are given by Greco *et al*. (2001b), by Dembczyński *et al*. (2003), and by Błaszczyński *et al*. (2010, 2011). In general, rule induction algorithms can be divided into three categories that reflect different induction strategies:

(α) generation of a minimal set of decision rules;
(β) generation of an exhaustive set of decision rules;
(γ) generation of a satisfactory set of decision rules.

When applied to a PCT, algorithms from category (α) focus on describing all pairs of objects from lower approximations of S and S^C by an m-set of m-rules. Algorithms from category (β) generate all m-rules. Category (γ) includes algorithms that generate all m-rules that satisfy some a priori defined requirements (concerning, e.g., maximum rule length or minimum support).

In this chapter, we apply the VC-DomLEM algorithm (Błaszczyński *et al*. 2010, 2011) which belongs to category (α). Each of the sets R_S and R_{S^C} of decision rules induced by VC-DomLEM for comprehensive preference relation S and S^C, respectively, is a c-m set of m-sc rules (i.e., a complete and minimal set composed of minimal and sufficiently consistent decision rules). Moreover, we parameterize the algorithm in such a way, that it induces d-rules [technically, this is achieved by choosing covering option $s = 1$, which means that each induced rule r_T is allowed to cover only pairs of objects belonging to set POS(T)]. It is important to note that the rules generated by VC-DomLEM do not have to be robust, which means that each rule r_T can employ elementary conditions created using evaluations in Q-evaluation space of different pairs of objects from $\underline{\mathrm{T}}$.

The next step of the proposed methodology to multicriteria ranking is the application of induced rules on set A. This application yields a specific preference structure on A. Each pair of objects $(a, b) \in A \times A$ can be covered by some decision rules suggesting assignment to relation S and/or to relation S^C. It can be also not covered by any rule. In order to address these possibilities, we define the following two crisp relations over

set A:

$$\mathcal{S} = \{(a, b) \in A \times A : (\exists\, r_S \in R_S \text{ such that } r_S \text{ covers } (a, b)) \text{ or } (aDb)\}, \tag{6.18}$$

$$\mathcal{S}^{\mathcal{C}} = \{(a, b) \in A \times A : (\exists\, r_{S^C} \in R_{S^C} \text{ such that } r_{S^C} \text{ covers } (a, b)) \text{ and not } (aDb)\}, \tag{6.19}$$

where $\exists\, r_T \in R_T$ is read as 'there exists a rule $r_T \in R_T$'. Let us observe that relation $\mathcal{S}$ is reflexive and relation $\mathcal{S}^{\mathcal{C}}$ is irreflexive. Moreover, relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$ are, in general, not transitive nor complete.

Both relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$ can be jointly represented by a *preference graph*. It is a directed multigraph $\mathcal{G}$. Each vertex (node) v_a of the preference graph corresponds to exactly one object $a \in A$. One can distinguish in $\mathcal{G}$ two types of arcs: $\mathcal{S}$-arcs and $\mathcal{S}^{\mathcal{C}}$-arcs. An $\mathcal{S}$-arc ($\mathcal{S}^{\mathcal{C}}$-arc) from vertex v_a to vertex v_b indicates that $a\mathcal{S}b$ (respectively, $a\mathcal{S}^{\mathcal{C}}b$). $\mathcal{G}$ is a multigraph since there may be one $\mathcal{S}$-arc and one $\mathcal{S}^{\mathcal{C}}$-arc for each pair of objects $(a, b) \in A \times A$. A *final recommendation* for the multicriteria ranking problem at hand, in terms of a complete or partial preorder of all objects belonging to set A, can be obtained upon a suitable exploitation of the preference graph.

6.6 Exploitation of preference graphs

The exploitation of preference graph $\mathcal{G}$, resulting from application of induced decision rules on set A, is not an easy task, especially because this graph represents two crisp relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$. This task is more complex than the exploitation of a preference graph representing just one crisp relation, well studied in the literature (see, e.g., Bouyssou and Vincke 1997; Vincke 1992).

Preference graphs representing only one crisp relation are obtained, e.g., in some decision support methods proposed in the field of MCDA, in which preferences of a DM are modeled in terms of binary relations. Among these methods, one can mention, e.g., ELECTRE IS (Figueira *et al*. 2005b; Roy and Skalka 1984). When preferences are modeled in terms of binary relations, the key question is the existence of evidence in favor of the considered relation. For example, in the case of outranking relation S concerned in the methods from the ELECTRE family, the evidence concerns the sentence aSb and/or bSa, for any pair of objects $a, b \in A$.

It is reasonable to claim that considering only evidence in favor of the considered binary relation does not allow to catch the reality of some decision problems. In fact, such an approach leads to a situation where the evidence in disfavor of a sentence is semantically considered – and thus modeled – as the evidence in favor of the opposite sentence. This mental restriction may induce not only misunderstandings but, which is even more important, it may also imply some loss of information [a good example clarifying this point, concerning government composition, is presented by Fortemps and Słowiński (2002)]. Therefore, in this chapter, given a pair of objects $(a, b) \in A \times A$, we consider not only the decision rules supporting conclusion aSb, but also the rules supporting the opposite conclusion, i.e., conclusion aS^Cb. In this way, we take into account the evidence in favor of preference of a over b and in disfavor of it; in the following, it will be also called positive evidence and negative evidence, respectively.

It is worth noting that the information contained in preference graph $\mathcal{G}$ can be represented using the *four-valued outranking* model of preferences, introduced by Tsoukias

and Vincke (1995, 1997); see also Greco *et al*. (1998). In this model, given a pair of objects $(a, b) \in A \times A$, one considers four possible situations for the outranking:

(1) *true outranking*, denoted by $a\mathcal{S}^{\mathcal{T}}b$, iff $a\mathcal{S}b$ and not $a\mathcal{S}^{\mathcal{C}}b$;

(2) *false outranking*, denoted by $a\mathcal{S}^{\mathcal{F}}b$, iff not $a\mathcal{S}b$ and $a\mathcal{S}^{\mathcal{C}}b$;

(3) *unknown outranking*, denoted by $a\mathcal{S}^{\mathcal{U}}b$, iff not $a\mathcal{S}b$ and not $a\mathcal{S}^{\mathcal{C}}b$;

(4) *contradictory outranking*, denoted by $a\mathcal{S}^{\mathcal{K}}b$, iff $a\mathcal{S}b$ and $a\mathcal{S}^{\mathcal{C}}b$.

The relations $\mathcal{S}^{\mathcal{T}}$, $\mathcal{S}^{\mathcal{F}}$, $\mathcal{S}^{\mathcal{U}}$, $\mathcal{S}^{\mathcal{K}}$, defined over A, correspond to the four truth values of Belnap (1976, 1977): T (true), F (false), U (unknown) and K (contradictory).

Given a preference graph $\mathcal{G}$, one can propose several exploitation techniques that lead to a final recommendation for the multicriteria ranking problem at hand, in terms of a complete or partial preorder of all objects from set A. We distinguish the following approaches:

(1) direct exploitation of preference relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$ using the *Net Flow Score* (NFS) procedure proposed by Greco *et al*. (1998);

(2) independent exploitation of preference relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$;

(3) suitable transformation of preference graph $\mathcal{G}$ to another graph $\tilde{\mathcal{G}}$ representing a *fuzzy relation* over set A, then exploitation of this relation leading to complete or partial preorder over A.

Before proceeding to the description of the three approaches to exploitation of preference graph $\mathcal{G}$, let us first introduce some concepts concerning fuzzy relations. Given a finite set of objects A, we denote by $\tilde{\mathbf{R}}_A$ the set of all fuzzy binary relations over A. Moreover, we use symbol $\tilde{R}$ to denote *any* fuzzy relation belonging to set $\tilde{\mathbf{R}}_A$. The membership of a pair of objects $(a, b) \in A \times A$ to relation $\tilde{R}$ is described by function $\mu_{\tilde{R}} : A \times A \to [0, 1]$. If $\mu_{\tilde{R}}(a, b) > 0$, pair (a, b) belongs to relation $\tilde{R}$ in degree $\mu_{\tilde{R}}(a, b)$. In case of $\mu_{\tilde{R}}(a, b) = 0$, pair (a, b) does not belong to relation $\tilde{R}$. In the following, value $\mu_{\tilde{R}}(a, b)$ is denoted in short by $\tilde{R}(a, b)$. Note that in the particular case when $\tilde{R}$ is crisp, $\tilde{R}(a, b) \in \{0, 1\}$, for any $a, b \in A$.

Approach (1) is based on scoring function $S^{NF} : A \to \Re$ defined as:

$$S^{NF}(a) = \sum_{b \in A \setminus \{a\}} \mathcal{S}(a, b) - \mathcal{S}(b, a) - \mathcal{S}^{\mathcal{C}}(a, b) + \mathcal{S}^{\mathcal{C}}(b, a). \tag{6.20}$$

Function S^{NF} induces a weak order over A, which is a solution of the considered multicriteria ranking problem. The idea of approach (2) is to exploit relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$ independently, obtaining two separate preorders (complete or partial), and then to conjunct these preorders in the same way as in the ELECTRE III method (Figueira *et al*. 2005b; Roy 1978). This leads to obtaining a partial preorder over A.

In the following, we will concentrate on approach (3), mainly for three reasons. The first one is that the exploitation of a fuzzy relation over a set of objects is well studied in the literature (Barrett *et al*. 1990; Bouyssou 1992a,b,c, 1995; Bouyssou and Perny 1992; Bouyssou and Pirlot 1997; Bouyssou and Vincke 1997; Dias and Lamboray 2010; Perny and Roy 1992; Pirlot 1995). Many so-called *ranking methods* have been proposed in this

subject. Greco *et al*. (1998) define a ranking method (RM) $\succeq$ as a function assigning a weak order $\succeq(A, \tilde{R})$ over A to any finite set A and any fuzzy relation $\tilde{R}$ over this set. We extend their definition as we define a ranking method $\succeq$ as a function assigning a partial preorder $\succeq(A, \tilde{R})$ over A to any finite set A and any fuzzy relation $\tilde{R}$ over this set. The diversity of ranking methods calls for a systematic comparison of their formal properties, which is, however, missing. The second reason for concentrating on approach (3) is that, using a suitable transformation of preference graph $\mathcal{G}$ and an appropriate ranking method to exploit the transformed graph $\tilde{\mathcal{G}}$, it is possible to obtain the same final ranking as in approach (1). Thus, approach (3) can be seen as a framework that encompasses approach (1). The third reason is that when applied to set A, most of the ranking methods considered in the literature yield a weak order over A, which is generally acknowledged to be more operational for a DM than a partial preorder that can be obtained in approach (2).

The suitable transformation of preference graph $\mathcal{G}$ representing two crisp preference relations $\mathcal{S}$ and $\mathcal{S}^{\mathcal{C}}$ to graph $\tilde{\mathcal{G}}$ representing one fuzzy relation $\tilde{\mathcal{R}} \in \tilde{\mathbf{R}}_A$ consists in defining relation $\tilde{\mathcal{R}}$ in the following way:

$$\tilde{\mathcal{R}}(a, b) = \frac{\mathcal{S}(a, b) + (1 - \mathcal{S}^{\mathcal{C}}(a, b))}{2}, \tag{6.21}$$

where $a, b \in A$. Such formulation of $\tilde{\mathcal{R}}$ is consistent with the scoring function S^{NF} defined as (6.20). Moreover, relation $\tilde{\mathcal{R}}$ is identical to the fuzzy relation over A considered by Greco *et al*. (1998), originally denoted by R_{4v}, defined as:

$$R_{4v}(a, b) = \begin{cases} 0 & \text{if } a\mathcal{S}^{\mathcal{F}}b \\ \frac{1}{2} & \text{if } a\mathcal{S}^{\mathcal{U}}b \text{ or } a\mathcal{S}^{\mathcal{K}}b \\ 1 & \text{if } a\mathcal{S}^{\mathcal{T}}b \end{cases},$$

where $a, b \in A$.

It is worth underlining that $\tilde{\mathcal{R}}(a, b) \in \{0, \frac{1}{2}, 1\}$, for any $(a, b) \in A \times A$. Therefore, we can call relation $\tilde{\mathcal{R}}$ as a *three-valued fuzzy relation*. Moreover, relation $\tilde{\mathcal{R}}$ is reflexive.

In the following, concerning the exploitation of relation $\tilde{\mathcal{R}}$, we assume that this relation has no 'structural properties' (Bouyssou 1996), i.e., we assume (what seems to be the case) that $\tilde{\mathcal{R}}$ may be *any* three-valued fuzzy relation over A. The rationale for this assumption is that relation $\tilde{\mathcal{R}}$ depends only on a considered set of decision rules, and, in general, this set of rules does not depend on A.

Let us introduce some useful concepts concerning exploitation of a fuzzy relation $\tilde{R}$ over a finite set of objects A. To this end, we adopt the notation of Bouyssou and Vincke (1997). First, we, respectively, denote by $=(A, \tilde{R})$ and $\succ(A, \tilde{R})$ the symmetric and asymmetric parts of $\succeq(A, \tilde{R})$, i.e., the relations such that, for all $a, b \in A$, ($a = (A, \tilde{R})\, b \Leftrightarrow a \succeq (A, \tilde{R})\, b$ and $b \succeq (A, \tilde{R})\, a$) and ($a \succ (A, \tilde{R})\, b \Leftrightarrow a \succeq (A, \tilde{R})\, b$ and not $b \succeq (A, \tilde{R})\, a$). Secondly, we denote by $\tilde{R}/A'$ the *restriction* of relation $\tilde{R}$ to set $A' \subseteq A$, i.e., fuzzy relation over A' such that for all $a, b \in A'$, $\tilde{R}/A'(a, b) = \tilde{R}(a, b)$. Thirdly, let $\tilde{R}$ be a crisp relation. We denote by $G(A, \tilde{R})$ the set of *greatest elements* of A given $\tilde{R}$, i.e., $G(A, \tilde{R}) = \{a \in A : \tilde{R}(a, b) = 1 \text{ for all } b \in A \backslash \{a\}\}$. It should be noticed that $G(A, \tilde{R})$ may well be empty. When $\tilde{R}$ is a weak order, it is easy to see that set $G(A, \tilde{R})$ is nonempty and equal to the first equivalence class of $\tilde{R}$.

In the literature, one can find many ranking methods 'dedicated' to exploitation of a fuzzy relation over a set of objects (Bouyssou 1992a,c; Bouyssou and Perny 1992;

Bouyssou and Pirlot 1997; Bouyssou and Vincke 1997; Pirlot 1995). On the other hand, as argued by Arrow and Raynaud (1986), one can be also interested in another approach to rank objects which consists in (downward) iterative application of a *choice function*. Let us denote by $\mathcal{P}_A$ the set of all nonempty subsets of a finite set of objects A. Then, choice function cf is a function

$$cf : \mathcal{P}_A \times \tilde{\mathbf{R}}_A \to \mathcal{P}_A. \tag{6.22}$$

A choice function associates with each nonempty set $A' \subseteq A$ and each fuzzy relation $\tilde{R}$ over A, a nonempty *choice set* $cf(A', \tilde{R}) \subseteq A'$, which may be interpreted as the set of the 'best' objects in A' given relation $\tilde{R}$. Iterative application of a choice function on a finite set A was considered, e.g., in Bouyssou (2004), Bouyssou and Pirlot (1997), Bouyssou and Vincke (1997). It leads to obtaining a weak order over A. Let us denote by $A^i \subseteq A$ the set of objects considered in the ith iteration and by $|A|$ the cardinality of set A. Obviously, $A^1 = A$. In the ith iteration, $i \in \{1, 2, \ldots, |A|\}$, given choice function cf is applied to set A^i. Then, the objects belonging to choice set $cf(A^i, \tilde{R})$ are put in the ith rank of the constructed ranking and removed from set A^i. Thus, $A^{i+1} = A^i \backslash cf(A^i, \tilde{R})$. The construction of a final ranking is finished when this ranking contains all objects from set A.

Most of the proposed 'dedicated' ranking methods as well as ranking methods based on iterative application of a choice function employ a *scoring function*. Given a finite set of objects A and a fuzzy relation $\tilde{R}$ over A, scoring function is used to evaluate relative performance of each object $a \in A$ w.r.t. the objects in nonempty set $A' \subseteq A$, taking into account relation $\tilde{R}$. Thus, scoring function sf is a function

$$sf : A \times \mathcal{P}_A \times \tilde{\mathbf{R}}_A \to \Re. \tag{6.23}$$

Value $sf(a, A', \tilde{R})$ denotes the *score* of object $a \in A$ calculated w.r.t. the objects in $A' \subseteq A$, given fuzzy relation $\tilde{R}$.

Let us define two *generic* score-based ranking methods: *single-stage ranking method* ($\succeq^1$) and *multi-stage ranking method* ($\succeq^i$). These ranking methods are parameterized by a set of objects A, a fuzzy relation $\tilde{R}$ over A, and a scoring function sf. Moreover, they yield a weak order over A:

- $\succeq^1(A, \tilde{R}, sf)$: assign score $sf(a, A, \tilde{R})$ to each object $a \in A$ and rank all the objects from set A according to their scores, in such a way that the higher the score of an object, the lower its rank (objects with the same score belong to the same rank);
- $\succeq^i(A, \tilde{R}, sf)$:
  - — define choice function cf as: $cf(A', \tilde{R}) = \{a \in A' : sf(a, A', \tilde{R}) \geq sf(b, A', \tilde{R}) \text{ for all } b \in A'\}$, where $A' \subseteq A$, i.e., in such a way that it chooses subset of A' composed of objects with the highest score;
  - — perform (downward) iterative application of choice function cf on set A.

Clearly, the aforementioned 'dedicated' ranking methods are *instances* of $\succeq^1$, differing only by the definition of function sf. Analogously, ranking methods based on iterative choice considered by Bouyssou and Pirlot (1997) and Bouyssou and Vincke (1997) are instances of $\succeq^i$, differing only by the definition of function sf.

Let us consider a finite set of objects A and a fuzzy relation $\tilde{R}$ over A. Then, according to Barrett *et al*. (1990), the score of any object $a \in A$ w.r.t. the objects in any set $A' \subseteq A$

can be calculated using one of the following scoring functions:

$$\textit{max in favor}: \quad MF(a, A', \tilde{R}) = \max_{b \in A' \setminus \{a\}} \tilde{R}(a, b), \tag{6.24}$$

$$\textit{min in favor}: \quad mF(a, A', \tilde{R}) = \min_{b \in A' \setminus \{a\}} \tilde{R}(a, b), \tag{6.25}$$

$$\textit{sum in favor}: \quad SF(a, A', \tilde{R}) = \sum_{b \in A' \setminus \{a\}} \tilde{R}(a, b), \tag{6.26}$$

$$\textit{−max against}: \quad -MA(a, A', \tilde{R}) = -\max_{b \in A' \setminus \{a\}} \tilde{R}(b, a), \tag{6.27}$$

$$\textit{−min against}: \quad -mA(a, A', \tilde{R}) = -\min_{b \in A' \setminus \{a\}} \tilde{R}(b, a), \tag{6.28}$$

$$\textit{−sum against}: \quad -SA(a, A', \tilde{R}) = -\sum_{b \in A' \setminus \{a\}} \tilde{R}(b, a), \tag{6.29}$$

$$\textit{max difference}: \quad MD(a, A', \tilde{R}) = \max_{b \in A' \setminus \{a\}} \tilde{R}(a, b) - \tilde{R}(b, a), \tag{6.30}$$

$$\textit{min difference}: \quad mD(a, A', \tilde{R}) = \min_{b \in A' \setminus \{a\}} \tilde{R}(a, b) - \tilde{R}(b, a), \tag{6.31}$$

$$\textit{sum of differences}: \quad SD(a, A', \tilde{R}) = \sum_{b \in A' \setminus \{a\}} \tilde{R}(a, b) - \tilde{R}(b, a). \tag{6.32}$$

It is worth noting that $SD(a, A', \tilde{R})$ is a sum of $SF(a, A', \tilde{R})$ and $-SA(a, A', \tilde{R})$.

Given a finite set of objects A and a fuzzy relation $\tilde{R}$ over A, we consider exploitation of relation $\tilde{R}$ using one of the following ranking methods, well studied in the literature:

(1) *Net Flow Rule* (Bouyssou 1992c; Bouyssou and Vincke 1997), defined as:

$$NFR(A, \tilde{R}) = \succeq^1(A, \tilde{R}, SD). \tag{6.33}$$

(2) *Iterative Net Flow Rule* (Bouyssou and Vincke, 1997), defined as:

$$It.NFR(A, \tilde{R}) = \succeq^i(A, \tilde{R}, SD). \tag{6.34}$$

(3) *Min in favor* (Bouyssou 1992a; Bouyssou and Pirlot 1997; Bouyssou and Vincke 1997; Pirlot 1995), defined as:

$$MiF(A, \tilde{R}) = \succeq^1(A, \tilde{R}, mF). \tag{6.35}$$

(4) *Iterative min in favor* (Bouyssou and Pirlot 1997), defined as:

$$It.MiF(A, \tilde{R}) = \succeq^i(A, \tilde{R}, mF). \tag{6.36}$$

(5) *Leaving and entering flows* (Bouyssou and Perny 1992), defined as:

$$L/E(A, \tilde{R}) = \succeq^1(A, \tilde{R}, SF) \quad \cap \quad \succeq^1(A, \tilde{R}, -SA). \tag{6.37}$$

As can be seen, the considered ranking methods employ only some of the defined scoring functions, namely: mF (6.25), SF (6.26), $-SA$ (6.29), and SD (6.32).

NFR orders objects according to their net flow scores. It has a long history in social choice theory (Arrow 1951; Fishburn, 1973). It coincides with the rule of Copeland (cf. Fishburn 1973; Henriet 1985; Rubinstein 1980) when $\tilde{R}$ is crisp. When $\tilde{R}(a, b)$ is interpreted as a percentage of voters considering that a is preferred or indifferent to b ($a, b \in A$), it corresponds to the well-known rule of Borda (cf. Fishburn 1973; Young 1974). Moreover, *NFR* is used in the PROMETHEE II outranking method (Brans and Mareschal 2005; Brans *et al*., 1984). *It.NFR* consists in iterative application of a choice function that chooses objects with the highest value of scoring function *SD* (6.32). This ranking method was originally called the *Repeated Net Flow Rule* and denoted by *RNFR* (Bouyssou and Vincke, 1997). L/E is used in the PROMETHEE I method (Brans et al., 1984; Brans and Mareschal, 2005). This ranking method allows any two objects $a, b \in A$ to be declared incomparable. This happens when two conclusions concerning ranking of these objects, one conclusion resulting only from the comparison of their *leaving flows*, i.e., values $SF(\cdot, A, \tilde{R})$, and the other one resulting only from the comparison of their *entering flows*, i.e., values $-(-SA(\cdot, A, \tilde{R}))$, are contradictory. Such contradiction occurs, e.g., when $SF(a, A, \tilde{R}) > SF(b, A, \tilde{R})$, while $-SA(a, A, \tilde{R}) < -SA(b, A, \tilde{R})$. It should be noticed that *NFR* and L/E make use of the 'cardinal' properties of values $\tilde{R}(a, b)$, with $a, b \in A$. On the other hand, *MiF* represents a prudent approach as it is purely 'ordinal' – it uses values $\tilde{R}(a, b)$ as if they were a numerical representation of a credibility of a crisp relation between a and b. Thus, from the fact that $\tilde{R}(a, b) \geq \tilde{R}(c, d)$ it concludes only that the relation between a and b is not less credible than the relation between c and d, with $a, b, c, d \in A$.

Now, let us go back and explain the text 'approach (3) can be seen as a framework that encompasses approach (1)', which appeared in the context of the four approaches for exploitation of preference graph $\mathcal{G}$. By saying this we meant that the weak order over A obtained using Equation (6.20) is the same as the weak order over A obtained using $NFR(A, \tilde{\mathcal{R}})$.

In the literature, one can find many properties considered in the context of ranking methods for fuzzy preference relations. These properties concern the result of application of a ranking method to any fuzzy relation or to a fuzzy relation with particular features, e.g., a relation which is crisp and transitive. It should be noticed, however, that these properties concern only the dependencies between the exploited fuzzy relation and obtained final ranking. In terms of our problem setup, this means that they do not concern the dependencies between comprehensive preference relations S, S^C and the final ranking.

The properties of ranking methods can be basically divided into two nondisjoint groups (Bouyssou 1992a; Bouyssou and Vincke 1997): desirable properties and 'characterizing' properties. The former reflect some expectations of a DM w.r.t. the final ranking produced by a ranking method. The latter reflect intrinsic characteristics of a ranking method; given a ranking method, the research concerning 'characterizing' properties aims at defining minimal sets of properties that a given ranking method is the only one to satisfy (Bouyssou 1992a,c; Bouyssou and Perny 1992; Bouyssou and Vincke 1997; Pirlot 1995). Since our goal is to obtain the 'best' ranking, we compare different ranking methods w.r.t. desirable properties only. The same route was taken, e.g., by Vincke (1992), in the context of exploitation of a crisp relation.

In general, different properties can be considered desirable in different decision problems (Bouyssou and Vincke 1997). We propose a list of properties that seem to be of interest for most decision problems. Moreover, in order to avoid a situation where all considered ranking methods become incomparable (nondominated), we suppose a priority order of considered desirable properties (which, from our point of view, reflects relative importance of these properties). This order is to be used only to resolve situations where two or more ranking methods satisfy the same maximum number of properties.

We find it reasonable to consider the following ordered list of desirable properties of a ranking method to be applied to exploitation of fuzzy relation $\tilde{\mathcal{R}}$ (6.21):

(1) *Neutrality* (property N)
This property was considered, e.g., by Bouyssou (1992a,c), Bouyssou and Perny (1992), Bouyssou and Vincke (1997), and Pirlot (1995).

Definition 6.6.1 (Neutrality) *A ranking method $\succeq$ is* neutral *if, for any finite set of objects A and any fuzzy relation $\tilde{R}$ over A:*
$(\sigma$ *is a permutation on* $A) \Rightarrow (a \succeq (A, \tilde{R})\, b \Leftrightarrow \sigma(a) \succeq (A, \tilde{R}^{\sigma})\, \sigma(b),$ *for all* $a, b \in A),$
where $\tilde{R}^{\sigma}$ is defined by $\tilde{R}^{\sigma}(\sigma(a), \sigma(b)) = \tilde{R}(a, b)$, for all $a, b \in A$.

Thus, neutrality expresses the fact that a ranking method does not discriminate between objects just because of their labels (or, in other words, their order in the considered set A). It is a classical property in this context (see, e.g., Henriet 1985; Rubinstein 1980).

(2) *Monotonicity* (property M)
Property of this name was considered, e.g., by Bouyssou and Perny (1992), Bouyssou and Vincke (1997), and Pirlot (1995), although the proposed definitions were semantically slightly different. In this chapter, we adopt the definition of monotonicity property given by Bouyssou and Perny (1992). Intuitively, monotonicity says that improving an object cannot decrease its position in the ranking and, moreover, deteriorating an object cannot improve its position in the ranking. In our opinion, the other two definitions previously considered miss at least one aspect of this intuitive formulation. Thus, we propose the following formulation of the monotonicity property.

Definition 6.6.2 Monotonicity *A ranking method $\succeq$ is* monotonic *if, for any finite set of objects A, any fuzzy relation $\tilde{R}$ over A, and any $a, b \in A$:*
$(a \succeq (A, \tilde{R})\, b \Rightarrow a \succeq (A, \tilde{R}')b),$
where $\tilde{R}'$ is identical to $\tilde{R}$ except that
$(\tilde{R}'(a, c) > \tilde{R}(a, c)$ *or* $\tilde{R}'(c, a) < \tilde{R}(c, a),$ *for some* $c \in A\backslash\{a\})$ *or*
$(\tilde{R}'(b, d) < \tilde{R}(b, d)$ *or* $\tilde{R}'(d, b) > \tilde{R}(d, b),$ *for some* $d \in A\backslash\{b\}).$

Precisely, the definition given by Pirlot (1995) w.r.t. the difference between $\tilde{R}'$ and $\tilde{R}$ concerns only that
$(\tilde{R}'(a, c) > \tilde{R}(a, c),$ for some $c \in A\backslash\{a\})$ or
$(\tilde{R}'(b, d) < \tilde{R}(b, d),$ for some $d \in A\backslash\{b\}).$

Moreover, the definition given by Bouyssou and Vincke (1997) lacks the second part of the above disjunction, i.e., the part concerning object b: $(\tilde{R}'(b,d) < \tilde{R}(b,d)$ or $\tilde{R}'(d,b) > \tilde{R}(d,b),$ for some $d \in A\backslash\{b\})$.

(3) *Covering compatibility* (property CC)
This property was considered, e.g., by Bouyssou and Vincke (1997) and Vincke (1992) (who called it *respect for the covering relation*).

Definition 6.6.3 (Covering compatibility) *A ranking method* $\succeq$ *is* covering compatible *if, for any finite set of objects A, any fuzzy relation* $\tilde{R}$ *over A, and any* $a, b \in A$: $(\tilde{R}(a,b) \geq \tilde{R}(b,a)$, *and for all* $c \in A\backslash\{a,b\}$, $\tilde{R}(a,c) \geq \tilde{R}(b,c)$ *and* $\tilde{R}(c,a) \leq \tilde{R}(c,b)) \Rightarrow a \succeq (A, \tilde{R})b$.

Thus, property CC expresses the intuition that when a 'covers' b, b should not be ranked before a. Our interest in this property results also from a very important fact – in the case of exploitation of fuzzy relation $\tilde{\mathcal{R}}$ defined by (6.21), property CC of applied ranking method guarantiees that the final ranking produced by this method respects dominance relation D over set A. Formally, this can be expressed by:

Corollary 6.6.4 *Given any two objects* $a, b \in A$, *such that aDb, property CC of ranking method* $\succeq$ *applied to exploitation of relation* $\tilde{\mathcal{R}}$ *(6.21) guarantiees that* $a \succeq (A, \tilde{\mathcal{R}})b$.

Proof. See the Appendix.

(4) *Discrimination* (property D)
This property was not considered in the literature concerning exploitation of a general fuzzy relation. However, it is important in the case of exploitation of a three-valued fuzzy relation $\tilde{\mathcal{R}}$ (6.21).

Definition 6.6.5 (Discrimination) *A ranking method* $\succeq$ *is* discriminatory *if for any finite set of objects A, there exists a fuzzy relation* $\tilde{R}$ *over A, such that the number of ranks in* $\succeq (A, \tilde{R})$ *is equal to the number of objects in set A.*

Thus, discrimination says that for each set of objects A there exists at least one fuzzy relation $\tilde{R}$ over A such that the ranking obtained by a considered ranking method is a complete order over set A.

(5) *Faithfulness* (property F)
This property was considered, e.g., by Bouyssou and Vincke (1997) and Vincke (1992) (who called it *respect for the data 1.1*).

Definition 6.6.6 (Faithfulness) *A ranking method* $\succeq$ *is* faithful *if, for any finite set of objects A and any relation* $\tilde{R}$ *over A:*
$(\tilde{R}$ *is a weak order over* $A) \Rightarrow (\succeq (A, \tilde{R}) = \tilde{R})$.

As can be seen, faithfulness concerns behavior of a ranking method in a special case when considered relation $\tilde{R}$ is crisp and, moreover, it is a weak order over A. This property says that a ranking method applied to a weak order should preserve it.

(6) *Data preservation* (property DP)
This property was considered, e.g., by Bouyssou and Vincke (1997) (where it was called *data-preservation 1*) and Vincke (1992) (who called it *respect for the data 1.3*).

Definition 6.6.7 (Data preservation) *A ranking method* $\succeq$ *is* data-preserving *if, for any finite set of objects A and any relation* $\tilde{R}$ *over A:*
($\tilde{R}$ *is a transitive crisp relation over* A) $\Rightarrow$ ($\tilde{R} \subseteq \succeq (A, \tilde{R})$).

Thus, data preservation says that when it is possible to obtain a partial preorder on the basis of $\tilde{R}$ without deleting information contained in this relation, a ranking method should do so. It is important to note that property DP is not implied by property F and vice versa.

(7) *Independence of nondiscriminating objects* (property $INDO$)
This property was considered, e.g., by Bouyssou and Vincke (1997) (where it was called *independence of nondiscriminating alternatives*) and by Vincke (1992) (who called it *independence of nondiscriminating elements: weak version*).

Definition 6.6.8 (Independence of nondiscriminating objects) *A ranking method* $\succeq$ *is* independent of nondiscriminating objects *if, for any finite set of objects A and any fuzzy relation* $\tilde{R}$ *over A:*
($\tilde{R}(a, b) = k$ *and* $\tilde{R}(b, a) = k^{'}$, *for all* $a \in A'$ *and all* $b \in A \backslash A'$, *with* $A' \subset A$) $\Rightarrow$ ($\succeq (A', \tilde{R}/A') = \succeq (A, \tilde{R})/A'$).

In the above definition, set $A \backslash A'$ is composed of nondiscriminating objects. Thus, independence of nondiscriminating objects says that when there is a subset of objects that compare in the same way with all other objects, the ranking of the other objects is not affected by the presence of this subset.

(8) *Independence of circuits* (property IC)
This property was considered, e.g., by Bouyssou (1992c) and Bouyssou and Vincke (1997). It reflects the way in which a ranking method deals with circuits (cycles) in the considered fuzzy relation. It uses the concept of *circuit equivalency* of two fuzzy relations.

Definition 6.6.9 (Circuit equivalency) *Let us consider a finite set of objects A. Two fuzzy relations* $\tilde{R}$ *and* $\tilde{R}^{'}$ *over A are* circuit-equivalent *if* $\tilde{R}^{'}$ *is identical to* $\tilde{R}$ *except that, for some distinct* $a, b, c \in A$ *and some* $\epsilon \in [-1, 1]$*:*
($\tilde{R}^{'}(a, b) = \tilde{R}(a, b) + \epsilon$ *and* $\tilde{R}^{'}(b, a) = \tilde{R}(b, a) + \epsilon$) *or*
($\tilde{R}^{'}(a, b) = \tilde{R}(a, b) + \epsilon$, $\tilde{R}^{'}(b, c) = \tilde{R}(b, c) + \epsilon$ *and* $\tilde{R}^{'}(c, a) = \tilde{R}(c, a) + \epsilon$).

Thus, $\tilde{R}^{'}$ and $\tilde{R}$ are circuit-equivalent if they are identical except for a circuit of length 2 or 3 on which a positive or negative value has been added.

Definition 6.6.10 (Independence of circuits) *A ranking method* $\succeq$ *is* independent of circuits *if, for any finite set of objects A and any two fuzzy relations* $\tilde{R}$ *and* $\tilde{R}^{'}$ *over A:*
($\tilde{R}^{'}$ *and* $\tilde{R}$ *are circuit-equivalent*) $\Rightarrow$ ($\succeq (A, \tilde{R}^{'}) = \succeq (A, \tilde{R})$).

According to Bouyssou and Vincke (1997), property *IC* has a straightforward interpretation. When $\tilde{R}'$ and $\tilde{R}$ are circuit-equivalent via a circuit of length 2, independence of circuits implies that the ranking is only influenced by the differences $\tilde{R}(a,b) - \tilde{R}(b,a)$. When $\tilde{R}'$ and $\tilde{R}$ are circuit-equivalent via a circuit of length 3, independence of circuits implies that intransitivities of the kind $\tilde{R}(a,b) > 0$, $\tilde{R}(b,c) > 0$ and $\tilde{R}(c,a) > 0$ can be 'wiped out'. It is important to notice that property *IC* makes an explicit use of the 'cardinal' properties of values $\tilde{R}(a,b)$, with $a, b \in A$ (except for the particular case in which both $\tilde{R}$ and $\tilde{R}'$ are crisp).

(9) *Ordinality* (property O)
This property was considered, e.g., by: Bouyssou (1992a), Bouyssou and Pirlot (1997), Bouyssou and Vincke (1997), and Pirlot (1995).

Definition 6.6.11 (Ordinality) *A ranking method* $\succeq$ *is* ordinal *if, for any finite set of objects A, any fuzzy relation* $\tilde{R}$ *over A, and any strictly increasing and one-to-one transformation* $\phi : [0,1] \to [0,1]$*:*
$\succeq (A, \phi[\tilde{R}]) = \ \succeq (A, \tilde{R})$,
where $\phi[\tilde{R}]$ *is the fuzzy relation on A such that* $\phi[\tilde{R}](a,b) = \phi(\tilde{R}(a,b))$, *for all* $a, b \in A$.

Thus, ordinality implies that a ranking method should not make use of the 'cardinal' properties of values $\tilde{R}(a,b)$, with $a, b \in A$.

(10) *Greatest faithfulness* (property GF)
This property was considered, e.g., by Bouyssou and Pirlot (1997) and Bouyssou and Vincke (1997).

Definition 6.6.12 (Greatest faithfulness) *A ranking method* $\succeq$ *is* greatest faithful *if, for any finite set of objects A and any relation* $\tilde{R}$ *over A:*
$\big(\tilde{R}\,\textit{is a crisp relation and}\; G(A,\tilde{R}) \neq \emptyset\big) \Rightarrow \big(G(A, \succeq (A,\tilde{R})) \subseteq G(A,\tilde{R})\big)$.

Greatest faithfulness says that if there are some greatest elements of a given set A, then the top-ranked objects should be chosen among them [observe that in case of a ranking method that yields a partial preorder over A, there may be no top-ranked objects, i.e., set $G(A, \succeq (A,\tilde{R}))$ may be empty]. Let us note, however, that some authors (e.g., Bouyssu and Pirlot 1997) do not find greatest faithfulness as a particularly intuitive requirement for a ranking method, as this property concerns only the first equivalence class of the obtained ranking (they rather consider this property in the context of choice methods). Moreover, in spite of names, it should be noticed that a faithful ranking method is not necessarily greatest faithful and vice versa.

Before verifying properties of the five considered ranking methods, let us make a note concerning reflexivity of an exploited fuzzy relation $\tilde{R} \in \tilde{\mathbf{R}}_A$. Vincke (1992) and Pirlot (1995) assumed $\tilde{R}$ to be irreflexive. Bouyssou (1992a), Bouyssou and Perny (1992) assumed that $\tilde{R}(a,b)$ is defined only for pairs of objects $(a,b) \in A \times A$ such that $a \neq b$. Finally, Bouyssou and Vincke (1997) and Bouyssou and Pirlot (1997) assumed that $\tilde{R}$ is reflexive. In this chapter, exploited relation $\tilde{\mathcal{R}}$ (6.21) is reflexive. However, since each of the five ranking methods analyzed here makes use of a scoring function that for any finite

Table 6.1 Properties of considered ranking methods for exploitation of a fuzzy relation.

Property/RM	*NFR*	*It.NFR*	*MiF*	*It.MiF*	L/E
N	**T**	**T**	**T**	*T*	*T*
M	*T*	*F*	*T*	*F*	**T**
CC	**T**	**T**	*T*	*T*	*T*
D	*T*	*T*	*F*	*T*	*T*
F	**T**	**T**	**F**	**T**	*T*
DP	**T**	**T**	*T*	*T*	*T*
$INDO$	**T**	**T**	*F*	*F*	*T*
IC	**T**	*F*	**F**	*F*	*F*
O	**F**	*F*	**T**	**T**	*F*
GF	**F**	*F*	**T**	**T**	*T*

set of objects A and any fuzzy relation $\tilde{R}$ over A does not take into account values $\tilde{R}(a, a)$, with $a \in A$, previous results concerning properties of the five ranking methods hold.

Table 6.1 presents properties of the five considered ranking methods. In this table, symbols T and F denote the presence and absence of a given property, respectively. Moreover, bold font is used in the case when a given pair (Property, RM) was already considered in the literature (where a proof or a counterexample was given), while italics is used otherwise, in which case a proof or a counterexample can be found in Szeląg *et al*. (2012). Note that in the row corresponding to property M, some symbols T and F are in italics due to adoption of particular definition of this property (see Definition 6.6.2).

Looking at Table 6.1, one can observe that the two ranking methods based on iterative application of a choice function, namely *It.NFR* and *It.MiF*, lack the monotonicity property. This observation is concordant with the work of Bouyssou (2004). Moreover, all ranking methods have property CC, which guarantiees that when they are applied to exploitation of fuzzy relation $\tilde{\mathcal{R}}$ (6.21), they produce final rankings respecting dominance relation D over set A.

Further analysis of the properties presented in Table 6.1 leads to the conclusion that, in view of the considered list of desirable properties, the best ranking method for exploitation of fuzzy relation $\tilde{\mathcal{R}}$ is the *NFR* method. This is because it satisfies most of the properties (which is, however, true also for the L/E ranking method) and, moreover, satisfies the first eight properties (i.e., N, M, CC, D, F, DP, $INDO$, and IC).

It is worth pointing out that the *NFR* ranking method is attractive not only because of the desirable properties it possesses. It represents an intuitive way of reasoning about relative worth of objects in set A, as it takes into account both positive and negative arguments concerning each object (i.e., strength and weakness of each object), as advocated by Fortemps and Słowiński (2002).

6.7 Illustrative example

Let us consider a hypothetical DM who is a scientist and wants to buy a notebook for personal use. The DM would like to spend no more than €1700. The DM is going to use

the notebook for writing scientific papers, programming, performing some computational experiments, and watching movies in his/her free time. For these reasons, the DM considers only 22 high-end notebooks, that have an Intel Core i7 processor with four cores, at least 4 MB of RAM (DDR3, 1333 MHz), and at least a 15 in. monitor with full high-definition resolution (1920 x 1080 pixels). The DM evaluates the notebooks using three cardinal criteria: price in euros (g_1, to be minimized), diagonal of a monitor in inches (g_2, to be maximized), and weight in kilograms (g_3, to be minimized). The weight is important because of the amount of work-related travel (e.g., attending conferences). The evaluations of all 22 notebooks using the three considered criteria are given in Table 6.2.

The set of objects from Table 6.2 constitutes set A of objects to be ranked. In the past, the DM has tested notebooks n_1, n_4, n_{10}, n_{12}, n_{14}, and n_{18} personally. These six objects constitute set A^R of reference objects. Based on personal experience, the DM is able to rank the six reference objects as follows: $n_4 \succ n_1 \succ n_{12} \succ n_{14} \succ n_{10} \succ n_{18}$ (i.e., object n_4 is the best, object n_1 is second best, ..., object n_{18} is the worst). Let us observe that the ranking of reference objects can be used as a source of preference information. Therefore, given any two notebooks $a, b \in A^R$, we fix aSb whenever notebook a is ranked by the DM not lower than notebook b. Moreover, we fix aS^Cb whenever notebook a is ranked lower than notebook b (in all such cases we have 'not aDb'). In this way, we get $B = A^R \times A^R$.

Table 6.2 Multicriteria evaluations of the considered notebooks.

Id	Model	Price (g_1)	Diagonal (g_2)	Weight (g_3)
n_1	Asus N75SF-V2G-TZ025V	865	17.3	3.4
n_2	Asus N75SF-V2G-TZ149V	877	17.3	3.4
n_3	Asus N75SL-V2G-TZ043V	1066	17.3	3.4
n_4	DELL XPS L502X	1031	15.6	2.7
n_5	Asus N55SL-S1072V	1042	15.6	2.84
n_6	Asus X93SM-YZ071V	971	18.4	4.11
n_7	DELL XPS 15	1372	15.6	2.51
n_8	DELL XPS L702X	1254	17.3	3.43
n_9	Samsung NP700G7A-S01PL	1656	17.3	3.2
n_{10}	Samsung NP700G7A-S02PL	1656	17.3	3.81
n_{11}	Asus G53SW-SZ141	1161	15.6	3.8
n_{12}	Asus G53SX-IX059V	1372	15.6	3.92
n_{13}	Asus G53SX-S1163V	1348	15.6	3.92
n_{14}	Asus G73SW-91037V	1538	17.3	3.9
n_{15}	Asus G74SX-TZ055V	1372	17.3	4.28
n_{16}	Asus G74SX-TZ210V	1419	17.3	4.28
n_{17}	Asus VX7SX-S1090V	1538	15.6	3.82
n_{18}	Lenovo ThinkPad T520	1467	15.6	2.5
n_{19}	Lenovo ThinkPad W520	1538	15.6	2.61
n_{20}	Sony VAIO VPC-F21Z1E	1467	16	3.1
n_{21}	Sony VAIO VPC-F23S1E	1419	16.4	3.1
n_{22}	Sony VAIO VPC-SE2V9E	1419	15.5	1.96

The data are taken from the internet store www.komputronik.pl.

Given the preference information, the following calculations are performed using jRank[1] software (Szeląg *et al*. 2010).

The preference information in the form of pairwise comparisons of six reference objects yields a PCT composed of 36 pairs of objects. This PCT is shown in Table 6.3. Let us note that the cardinality of relation S is 21, and the cardinality of relation S^C is 15.

One can observe in the PCT several inconsistencies w.r.t. dominance relation D_2 over set B. Such inconsistency occurs when a pair of objects $(a, b) \in S$ is dominated by a pair of objects $(c, d) \in S^C$. Inconsistent pairs of objects appearing in Table 6.3 are marked in the table by an asterisk. All inconsistencies w.r.t. dominance relation D_2 over B are also presented in Table 6.4, where an asterisk indicates where pair $(a, b) \in S$ from the corresponding row is inconsistent with pair $(c, d) \in S^C$ from the corresponding column.

In order to show potential advantage of VC-DRSA over DRSA, when applied to preference learning in multicriteria raking, we consider two independent calculation paths, taking the PCT shown in Table 6.3 as a 'point of departure'. They are composed of the following steps:

(s_1) calculation of lower approximations of relations S and S^C, according to definitions (6.18) and (6.19), respectively;

(s_2) calculation of a minimal set of decision rules by VC-DomLEM algorithm;

(s_3) application of the induced rules on set A;

(s_4) exploitation of the resulting preference structure by the *NFR* ranking method;

(s_5) evaluation of the obtained final ranking over set A.

Each of the above steps 'produces' some results. These are: lower approximations of outranking and non outranking relations obtained in step (s_1), sets R_S and R_{S^C} of minimal decision rules obtained in step (s_2), preference structure on A obtained in step (s_3), final ranking (weak order) on A obtained in step (s_4), and value of a chosen error measure obtained in step (s_5). However, these results differ in both calculation paths only due to decisions made in step (s_1), concerning consistency thresholds θ_S and θ_{S^C} used to define lower approximations (6.11) and (6.12). In the first path, denoted by $cp_{\theta=0}$, we assume that both consistency thresholds are equal to zero. Thus, calculated lower approximations are the same as in the case of DRSA (Greco *et al* 1999a, 2001a; Słowiński *et al*. 2005). In the second path, denoted by $cp_{\theta>0}$, we choose $\theta_S = \theta_{S^C} = 0.1$. In this way, we relax a little bit the conditions for inclusion of pairs of objects to lower approximations (6.11) and (6.12). In particular, a pair of objects $(a, b) \in S$ is considered to be sufficiently consistent (and thus included in $\underline{S}$) if it is dominated by at most one pair of objects belonging to relation S^C [this can be verified using Equation (6.9): $1/15 = 0.067 < \theta_S = 0.1 < 2/15 = 0.133$]. Moreover, a pair of objects $(a, b) \in S^C$ is considered to be sufficiently consistent (and thus included in $\underline{S^C}$) if it dominates at most two pairs of objects belonging to relation S [this can be verified using Equation (6.10): $2/21 = 0.095 < \theta_{S^C} = 0.1 < 3/21 = 0.143$].

In step (s_5), we take into account dependencies (6.1), (6.2), and (6.3), to represent the pairwise comparisons of the reference objects in terms of relations P, P^{-1}, and I.

[1] See http://www.cs.put.poznan.pl/mszelag/Software/jRank/jRank.html.

Table 6.3 The PCT yielded by pairwise comparisons of six reference objects.

(a, b)	Δ_1	Δ_2	Δ_3	Relation
(n_4, n_4)	0	0.0	0.0	S
$(n_4, n_1)^*$	166	−1.7	−0.7	S
(n_4, n_{12})	−341	0.0	−1.22	S
(n_4, n_{14})	−507	−1.7	−1.2	S
(n_4, n_{10})	−625	−1.7	−1.11	S
(n_4, n_{18})	−436	0.0	0.2	S
$(n_1, n_4)^*$	−166	1.7	0.7	S^C
(n_1, n_1)	0	0.0	0.0	S
(n_1, n_{12})	−507	1.7	−0.52	S
(n_1, n_{14})	−673	0.0	−0.5	S
(n_1, n_{10})	−791	0.0	−0.41	S
(n_1, n_{18})	−602	1.7	0.9	S
(n_{12}, n_4)	341	0.0	1.22	S^C
(n_{12}, n_1)	507	−1.7	0.52	S^C
(n_{12}, n_{12})	0	0.0	0.0	S
$(n_{12}, n_{14})^*$	−166	−1.7	0.02	S
(n_{12}, n_{10})	−284	−1.7	0.11	S
$(n_{12}, n_{18})^*$	−95	0.0	1.42	S
(n_{14}, n_4)	507	1.7	1.2	S^C
(n_{14}, n_1)	673	0.0	0.5	S^C
$(n_{14}, n_{12})^*$	166	1.7	−0.02	S^C
(n_{14}, n_{14})	0	0.0	0.0	S
(n_{14}, n_{10})	−118	0.0	0.09	S
$(n_{14}, n_{18})^*$	71	1.7	1.4	S
(n_{10}, n_4)	625	1.7	1.11	S^C
(n_{10}, n_1)	791	0.0	0.41	S^C
(n_{10}, n_{12})	284	1.7	−0.11	S^C
(n_{10}, n_{14})	118	0.0	−0.09	S^C
(n_{10}, n_{10})	0	0.0	0.0	S
$(n_{10}, n_{18})^*$	189	1.7	1.31	S
(n_{18}, n_4)	436	0.0	−0.2	S^C
(n_{18}, n_1)	602	−1.7	−0.9	S^C
$(n_{18}, n_{12})^*$	95	0.0	−1.42	S^C
$(n_{18}, n_{14})^*$	−71	−1.7	−1.4	S^C
$(n_{18}, n_{10})^*$	−189	−1.7	−1.31	S^C
(n_{18}, n_{18})	0	0.0	0.0	S

Pairs of objects marked by an asterisk are inconsistent.

Moreover, we use the following representation of a final ranking $\succeq (A, \tilde{\mathcal{R}})$ in terms of relations $P_{\succeq}$, $P_{\succeq}^{-1}$, and $I_{\succeq}$:

$$aP_{\succeq}b \Leftrightarrow a \succeq (A, \tilde{\mathcal{R}})\ b \text{ and not } b \succeq (A, \tilde{\mathcal{R}})\ a, \tag{6.38}$$

$$aP_{\succeq}^{-1}b \Leftrightarrow \text{ not } a \succeq (A, \tilde{\mathcal{R}})\ b \text{ and } b \succeq (A, \tilde{\mathcal{R}})\ a, \tag{6.39}$$

$$aI_{\succeq}b \Leftrightarrow a \succeq (A, \tilde{\mathcal{R}})\ b \text{ and } b \succeq (A, \tilde{\mathcal{R}})\ a, \tag{6.40}$$

Table 6.4 Inconsistencies in the PCT yielded by pairwise comparisons of six reference objects.

	(n_1, n_4)	(n_{14}, n_{12})	(n_{18}, n_{12})	(n_{18}, n_{14})	(n_{18}, n_{10})
(n_4, n_1)			*	*	*
(n_{12}, n_{14})					*
(n_{12}, n_{18})	*				
(n_{14}, n_{18})	*				
(n_{10}, n_{18})	*	*			

The asterisk indicates where pair $(a, b) \in S$ from the corresponding row is inconsistent with pair $(c, d) \in S^C$ from the corresponding column.

where $a, b \in A$. Thus, $aP_{\succeq}b$ iff object a is ranked higher than object b, $aP_{\succeq}^{-1}b$ iff object a is ranked lower than object b, and $aI_{\succeq}b$ iff the ranks of objects a and b are equal. Then, taking advantage of the fact that $A^R \subseteq A$, we apply *Kendall rank correlation coefficient* τ to measure the concordance between a final ranking of all 22 objects and the pairwise comparisons of the six reference objects (derived from the 'short' ranking given by the DM).

Kendall rank correlation coefficient τ, applied to measure concordance of a ranking $\succeq (A, \tilde{\mathcal{R}})$, represented in terms of relations $P_{\succeq}$, $P_{\succeq}^{-1}$, and $I_{\succeq}$, w.r.t. given pairwise comparisons of reference objects represented in terms of relations P, P^{-1}, and I, is defined as:

$$\tau(P_{\succeq}, P_{\succeq}^{-1}, I_{\succeq}, P, P^{-1}, I) = 1 - 2\frac{\sum\limits_{(a,b)\in B, a\neq b} err(a, b)}{|\{(a, b) \in B : a \neq b\}|}, \tag{6.41}$$

where $err(a, b)$ denotes an error accounted for a pair of objects $(a, b) \in B$, $a \neq b$. This error is defined as:

$$err(a, b) = \begin{cases} 0, & \text{if } (aPb \text{ and } aP_{\succeq}b) \text{ or } (aP^{-1}b \text{ and } aP_{\succeq}^{-1}b) \text{ or } (aIb \text{ and } aI_{\succeq}b) \\ \frac{1}{2}, & \text{if } \big((aPb \text{ or } aP^{-1}b) \text{ and } aI_{\succeq}b\big) \text{ or } (\ aIb \text{ and } (aP_{\succeq}b \text{ or } aP_{\succeq}^{-1}b) \\ 1, & \text{if } (aPb \text{ and } aP_{\succeq}^{-1}b) \text{ or } (aP^{-1}b \text{ and } aP_{\succeq}b) \end{cases}. \tag{6.42}$$

Values of coefficient τ belong to the interval $[-1, 1]$. The best possible value of τ is 1, and the worst possible value is -1.

Table 6.5 summarizes the results obtained in subsequent steps (s_1)–(s_5), along both calculation paths.

Looking at Table 6.5, it is clear that the results obtained along calculation path $cp_{\theta>0}$ are better (greater lower approximations and higher value of τ). Thus, for the considered

Table 6.5 Summary of results obtained in steps $(s_1) - (s_5)$, for calculations paths $cp_{\theta=0}$, $cp_{\theta>0}$.

Calculation path	$\lvert\underline{S}\rvert$	$\lvert\underline{S^C}\rvert$	$\lvert R_S\rvert$	$\lvert R_{S^C}\rvert$	τ
$cp_{\theta=0}$	16	10	3	2	0.400
$cp_{\theta>0}$	19	14	3	2	0.733

Table 6.6 Minimal decision rules induced by the VC-DomLEM algorithm.

Decision rule r_T	Supp	$\widehat{\epsilon}_T(r_T)$
if $\big(\Delta_1(a,b) \leq -284\big)$, *then aSb*	9	0
if $\big(\Delta_1(a,b) \leq -166\big) \wedge \big(\Delta_3(a,b) \leq 0.02\big)$, *then aSb*	7	0.067
if $\big(\Delta_1(a,b) \leq 71\big) \wedge \big(\Delta_2(a,b) \geq 0\big)$, *then aSb*	15	0.067
if $\big(\Delta_1(a,b) \geq 95\big)$, *then* aS^Cb	12	0.095
if $\big(\Delta_1(a,b) \geq -189\big) \wedge \big(\Delta_2(a,b) \leq -1.7\big)$, *then* aS^Cb	14	0.095

'Supp' denotes the number of pairs of objects that support given rule r_T.

illustrative example, VC-DRSA proved to be more useful than DRSA. In view of this conclusion, in the following, we present only results obtained along calculation path $cp_{\theta>0}$.

The set of minimal decision rules induced by the VC-DomLEM algorithm is presented in Table 6.6, where 'supp' denotes the number of pairs of objects that support given rule r_T. Note that the third rule covers all pairs of objects $(a, b) \in A \times A$ such that aDb, and that none of the rules suggesting assignment to relation S^C covers any such pair of objects. It is worth noting that the induced rules are relatively short and the number of rules is small w.r.t. the size of the PCT. Moreover, the rules are easy to interpret by the DM.

The final ranking of all objects from set A, obtained using the *NFR* ranking method, is presented in Table 6.7, where the six reference objects are marked in bold, and for each rank we give the respective net flow score, i.e., the value of scoring function S^{NF} (6.20).

Table 6.7 Final ranking of all objects from set A, obtained using the *NFR* ranking method.

Rank	Net flow score	Object(s)
1	39.0	$\mathbf{n}_1$
2	38.0	n_2
3	37.0	n_6
4	30.0	n_3
5	24.0	$\mathbf{n}_4$, n_5
6	17.0	n_8
7	14.0	n_{11}
8	7.0	n_{15}
9	6.0	n_{16}
10	−2.0	n_{21}
11	−7.0	n_{13}
12	−8.0	n_7, $\mathbf{n}_{12}$
13	−14.0	n_{14}
14	−16.0	n_{20}
15	−20.0	n_{22}
16	−27.0	$\mathbf{n}_{18}$
17	−32.0	n_{17}, n_{19}
18	−35.0	n_9, $\mathbf{n}_{10}$

Bold indicates a reference object.

Now, let us analyze how we obtained $\tau = 0.733$. According to (6.42), the error value is equal to one in four cases since $n_4 P n_1$ but $n_4 P_{\succeq}^{-1} n_1$, $n_{10} P n_{18}$ but $n_{10} P_{\succeq}^{-1} n_{18}$, $n_1 P^{-1} n_4$ but $n_1 P_{\succeq} n_4$, and $n_{18} P^{-1} n_{10}$ but $n_{18} P_{\succeq} n_{10}$. Third, $|\{(a, b) \in B : a \neq b\}| = 30$. Thus, from (6.41) we get $\tau = 1 - 2 * \frac{4}{30} = 0.733$.

It is worth underlining that the pairs of objects that got 'inverted' (i.e., changed relation from P to $P_{\succeq}^{-1}$, or from P^{-1} to $P_{\succeq}$) were inconsistent (see Table 6.4). Taking this into account, we can say that the induced set of rules is a good preference model of the DM.

6.8 Summary and conclusions

We presented a methodology for dealing with multicriteria ranking problems. It concerns construction of a comprehensible preference model in the form of a set of '*if* ..., *then* ...' decision rules. This model is constructed by generalization of decision examples supplied by a DM in the form of pairwise comparisons of some reference objects.

We considered a situation where for any two different reference objects a, b, that the DM wishes to compare, he/she can declare that either 'object a is at least as good as object b' or 'object a is not at least as good as object b'. In this way, the DM specifies two comprehensive preference relations. Such preference information is used to create a PCT. As this information is prone to inconsistencies, we proposed to structure it using VC-DRSA, by calculation of lower approximations of the two comprehensive preference relations. In this way, we can restrict a priori the set of pairs of objects which serves as a basis for induction of decision rules to a subset of sufficiently consistent pairs of objects. This restriction is motivated by the goal of learning a 'reliable' preference model.

Application of induced decision rules on the whole set of objects to be ranked, yields a specific preference structure (directed graph) in this set. This preference structure is then exploited using a ranking method, so as to work out a final recommendation, i.e., a ranking of objects. We proposed a list of some desirable properties that a ranking method is expected to have, and compared several ranking methods studied in the literature w.r.t. these properties. Based on this comparison, we chose one ranking method, the NFR, which has the best properties.

We illustrated the proposed methodology by an example concerning ranking of notebooks. This example shows usefulness of the proposed methodology for dealing with multicriteria ranking problems.

Acknowledgment

The Polish authors wish to acknowledge financial support from the Polish National Science Centre, grant no. N N519 441939.

References

Arrow K (1951) *Social Choice and Individual Values*. Yale University Press New Haven, CT.

Arrow K and Raynaud H (1986) *Social Choice and Multicriterion Decision Making*. MIT Press, Cambridge.

Barrett C, Pattanaik PK and Salles M (1990) On choosing rationally when preferences are fuzzy. *Fuzzy Sets and Systems* **34**, 197–212.

Belnap N (1976) How a computer should think. In *Proceedings of the Oxford International Symposium on Contemporary Aspects of Philosophy* (ed. Ryle G). Oriel Press, Stocksfield, pp. 30–56.

Belnap N (1977) A useful four-valued logic. In *Modern Uses of Multiple Valued Logics* (eds Dunn JM and Epstein G). D. Reidel, Dordrecht, pp. 8–37.

Błaszczyński J, Greco S and Słowiński R (2012) Inductive discovery of laws using monotonic rules. *Engineering Applications of Artificial Intelligence* **25**(2), 284–294.

Błaszczyński J, Greco S, Słowiński R and Szeląg M (2006) On variable consistency dominance-based rough set approaches. In *Rough Sets and Current Trends in Computing 2006* (eds Greco S, Hata Y, Hirano S, Inuiguchi M, Miyamoto S, Nguyen HS and Słowiński R), vol. 4259 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pp. 191–202.

Błaszczyński J, Greco S, Słowiński R and Szeląg M (2007) Monotonic variable consistency rough set approaches. In *Rough Sets and Knowledge Technology 2007* (eds Yao J, Lingras P, Wu W, Szczuka M, Cercone NJ and Ślęzak D), vol. 4481 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pp. 126–133.

Błaszczyński J, Greco S, Słowiński R and Szeląg M (2009) Monotonic variable consistency rough set approaches. *International Journal of Approximate Reasoning* **50**(7), 979–999.

Błaszczyński J, Słowiński R and Szeląg M (2010) Probabilistic rough set approaches to ordinal classification with monotonicity constraints. In *IPMU 2010* (eds Hüllermeier E, Kruse R and Hoffmann F), vol. 6178 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pp. 99–108.

Błaszczyński J, Słowiński R and Szeląg M (2011) Sequential covering rule induction algorithm for variable consistency rough set approaches. *Information Sciences* **181**, 987–1002. Available online 14 November 2010.

Bouyssou D (1992a) A note on the 'min in favor' ranking method for valued preference relations. In *Multicriteria Decision Making. Methods – Algorithms – Applications* (eds Cerny M, Glükaufová D and Loula D). Czechoslovak Academy of Sciences, Prague, pp. 16–25.

Bouyssou D (1992b) A note on the sum of differences choice function for fuzzy preference relations. *Fuzzy Sets and Systems* **47**, 197–202.

Bouyssou D (1992c) Ranking methods based on valued preference relations: A characterization of the net flow method. *European Journal of Operational Research* **60**, 61–67.

Bouyssou D (1995) A note on the 'min in favor' choice procedure for fuzzy preference relations. In *Advances in Multicriteria Analysis* (eds Pardalos P, Siskos Y and Zopounidis C). Kluwer, Dordrecht, pp. 9–16.

Bouyssou D (1996) Outranking relations: do they have special properties?. *Journal of Multi-Criteria Decision Analysis* **5**, 99–111.

Bouyssou D (2004) Monotonicity of 'ranking by choosing': A progress report. *Social Choice and Welfare* **23**, 249–273.

Bouyssou D and Perny P (1992) Ranking methods for valued preference relations: A characterization of a method based on leaving and entering flows. *European Journal of Operational Research* **61**, 186–194.

Bouyssou D and Pirlot M (1997) Choosing and ranking on the basis of fuzzy preference relations with the 'min in favour'. In *Multiple Criteria Decision Making - Proceedings of the Twelfth International Conference Hagen (Germany)* (eds Fandel G and Gal T). Springer-Verlag Berlin, pp. 115–127.

Bouyssou D and Vincke P (1997) Ranking alternatives on the basis of preference relations: A progress report with special emphasis on outranking relations. *Journal of Multi-Criteria Decision Analysis* **6**, 77–85.

Brans J and Mareschal B (2005) PROMETHEE methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M), Operations Research & Management Science. Springer-Verlag, New York, pp. 163–196.

Brans J, Mareschal B and Vincke P (1984) Promethee: a new family of outranking methods in multicriteria analysis. In *Operational Research, IFORS 84* (ed. Brans J). North Holland, Amsterdam, pp. 477–490.

Dembczyński K, Pindur R and Susmaga R (2003) Generation of exhaustive set of rules within dominance-based rough set approach. *Electronic Notes in Theoretical Computer Science* **82**(4), 96–107.

Dias LC and Lamboray C (2010) Extensions of the prudence principle to exploit a valued outranking relation. *European Journal of Operational Research* **201**(3), 828–837.

Figueira J, Greco S and Ehrgott M (eds) (2005a) *Multiple Criteria Decision Analysis: State of the Art Surveys*, Operations Research & Management Science. Springer-Verlag, New York.

Figueira J, Greco S and Słowiński R (2009) Building a set of additive value functions representing a reference preorder and intensities of preference: Grip method. *European Journal of Operational Research* **195**(2), 460–486.

Figueira J, Mousseau V and Roy B (2005b) ELECTRE methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (ed. Figueira J, Greco S and Ehrgott M), Operations Research & Management Science. Springer-Verlag, New York, pp. 133–162.

Fishburn P (1973) *The Theory of Social Choice*. Princeton University Press, Princeton, NJ.

Fortemps P, Greco S and Słowiński R (2008) Multicriteria decision support using rules that represent rough-graded preference relations. *European Journal of Operational Research* **188**(1), 206–223.

Fortemps P and Słowiński R (2002) A graded quadrivalent logic for ordinal preference modelling: Loyola-like approach. *Fuzzy Optimization and Decision Making* **1**(1), 93–111.

Giove S, Greco S, Matarazzo B and Słowiński R (2002) Variable consistency monotonic decision trees. In *Rough Sets and Current Trends in Computing (RSCTC 2002)* (eds Alpigini JJ, Peters JF, Skowron A and Zhong N), vol. 2475 of *Lecture Notes in Artificial Intelligence*. Springer, Malvern, PA, pp. 247–254.

Greco S, Kadziński M, Mousseau V and Słowiński R (2011) Electre^{GKMS}: Robust ordinal regression for outranking methods. *European Journal of Operational Research* **214**(10), 118–135.

Greco S, Matarazzo B and Słowiński R (1995) Rough set approach to multi-attribute choice and ranking problems. Technical Report 38/95, ICS, Warsaw University of Technology, Warsaw.

Greco S, Matarazzo B and Słowiński R (1997) Rough set approach to multi-attribute choice and ranking problems. In *Multiple Criteria Decision Making, Proceedings of the Twelfth International Conference, Hagen, Germany* (eds Fandel G and Gal T). Springer, Berlin, pp. 318–329.

Greco S, Matarazzo B and Słowiński R (1999a) Rough approximation of a preference relation by dominance relations. *European Journal of Operational Research* **117**, 63–83.

Greco S, Matarazzo B and Słowiński R (1999b) The use of rough sets and fuzzy sets in MCDM. In *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications* (eds Gal T, Stewart TJ and Hanne T), vol. 21 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Dordrecht, chapter 14, pp. 14.1–14.59.

Greco S, Matarazzo B and Słowiński R (2001a) Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research* **129**(1), 1–47.

Greco S, Matarazzo B and Sł owiński R (2005) Decision rule approach. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M), Operations Research & Management Science. Springer-Verlag, New York, chapter 13, pp. 507–562.

Greco S, Matarazzo B and Słowiński R (2008a) Dominance-based rough set approach to interactive multiobjective optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches* (eds Branke J, Deb K, Miettinen K and Słowiński R), vol. 5252 of *Lecture Notes in Computer Science*. Springer, Berlin, chapter 5, pp. 121–156.

Greco S, Matarazzo B and Słowiński R (2010a) Dominance-based rough set approach to decision under uncertainty and time preference. *Annals of Operations Research* **176**(1), 41–75.

Greco S, Matarazzo B, Słowiński R and Tsoukiàs A (1998) Exploitation of a rough approximation of the outranking relation in multicriteria choice and ranking. In *Trends in Multicriteria Decision Making* (eds Stewart TJ and van den Honert RC), vol. 465 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, pp. 45–60.

Greco S, Mousseau V and Słowiński R (2008b) Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research* **191**(2), 415–435.

Greco S, Pawlak Z and Słowiński R (2004) Can bayesian confirmation measures be useful for rough set decision rules?. *Engineering Applications of Artificial Intelligence* **17**, 345–361.

Greco S, Słowiński R, Figueira J and Mousseau V (2010b) Robust ordinal regression. In *Trends in Multiple Criteria Decision Analysis* (eds Ehrgott M, Figueira J and Greco S). Springer, New York, pp. 241–283.

Greco S, Słowiński R and Matarazzo B (2001b) An algorithm for induction of decision rules consistent with dominance principle. In *Rough Sets and Current Trends in Computing 2001* (eds Ziarko W and Yao YY), vol. 2005 of *Lecture Notes in Artificial Intelligence*. Springler-Verlag, Berlin, pp. 304–313.

Greco S, Słowiński R, Matarazzo B and Stefanowski J (2001c) Variable consistency model of dominance-based rough sets approach. In *Rough Sets and Current Trends in Computing 2001* (eds Ziarko W and Yao YY), vol. 2005 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pp. 170–181.

Henriet D (1985) The copeland choice function – an axiomatic characterization. *Social Choice and Welfare* **2**, 49–64.

Inuiguchi M and Yoshioka Y (2006) Variable-precision dominance-based rough set approach. In *Rough Sets and Current Trends in Computing 2006* (eds Greco S, Hata Y, Hirano S, Inuiguchi M, Miyamoto S, Nguyen HS and Słowiński R), vol. 4259 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, pp. 203–212.

Langley P and Simon HA (1998) Fielded applications of machine learning. In *Machine Learning and Data Mining* (eds Michalski RS, Bratko I and Kubat M). John Wiley & Sons, Ltd, New York pp. 113–129.

March JG (1988) Bounded rationality, ambiguity, and the engineering of choice. In *Decision Making, Descriptive, Normative and Prescriptive Interactions* (eds Bell DE, Raiffa H and Tversky A). Cambridge University Press, New York, pp. 33–58.

Michalski RS (1983) A theory and methodology of inductive learning. In *Machine Learning: An Artificial Intelligence Approach* (eds Michalski RS, Carbonell JG and Mitchell TM). Tioga Publishing, Palo Alto, pp. 83–129.

Mousseau V and Słowiński R (1998) Inferring an ELECTRE TRI model from assignment examples. *Journal of Global Optimization* **12**(2), 157–174.

Pawlak Z (1991) *Rough Sets. Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht.

Pawlak Z and Słowiński R (1994) Rough set approach to multi-attribute decision analysis. *European Journal of Operational Research* **72**(3), 443–459.

Perny P and Roy B (1992) The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems* **49**, 33–53.

Pirlot M (1995) A characterization of 'min' as a procedure for exploiting valued preference relations and related results. *Journal of Multi-Criteria Decision Analysis* **4**, 37–56.

Roy B (1978) ELECTRE III : Un algorithme de classements fondé sur une représentation floue des préférences en présence de critères multiples. *Cahiers du CERO* **20**(1), 3–24.

Roy B (1991) The outranking approach and the foundation of ELECTRE methods. *Theory and Decision* **31**, 49–73.

Roy B (1996) *Multicriteria Methodology for Decision Aiding*. Kluwer, Dordrecht.

Roy B and Bouyssou D (1993) *Aide Multicritère à la Décision: Méthodes et Cas*. Economica, Paris.

Roy B and Skalka JM (1984) ELECTRE IS: Aspects méthodologiques et guide d'utilisation. *Document du LAMSADE* no. 30, Université Paris-Dauphine, Paris, France.

Rubinstein A (1980) Ranking the participants in a tournament. *SIAM Journal of Applied Mathematics* **38**, 108–111.

Slovic P (1975) Choice between equally-valued alternatives. *Journal of Experimental Psychology: Human Perception Performance* **1**(3), 280–287.

Słowiński R (1993) Rough set learning of preferential attitude in multi-criteria decision making. In *Methodologies for Intelligent Systems* (eds Komorowski J and Raś Z), vol. 689 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin, pp. 642–651.

Słowiński R, Greco S and Matarazzo B (2005) Rough set based decision support. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (eds Burke EK and Kendall G). Springer-Verlag, New York, chapter 16, pp. 475–527.

Słowiński R, Greco S and Matarazzo B (2009) Rough sets in decision making. In *Encyclopedia of Complexity and Systems Science* (ed. Meyers R). Springer-Verlag, New York, pp. 7753–7786.

Szeląg M, Greco S and Słowiński R (2012) Variable-consistency dominance-based rough set approach to multicriteria ranking. Research Report RA-02/12, Poznań University of Technology.

Szeląg M, Słowiński R and Błaszczyński J (2010) jRank – ranking using dominance-based rough set approach. *Newsletter of the European Working Group 'Multiple Criteria Decision Aiding'* **3**(22), 13–15.

Tsoukias A and Vincke P (1995) A new axiomatic foundation of partial comparability. *Theory and Decision* **39**, 79–114.

Tsoukias A and Vincke P (1997) A new axiomatic foundation of the partial comparability theory. In *Multicriteria Analysis* (ed. Climaco J). Springer-Verlag, Berlin, pp. 37–50.

Vincke P (1992) Exploitation of a crisp relation in a ranking problem. *Theory and Decision* **32**, 221–240.

Young H (1974) An axiomatization of Borda's rule. *Journal of Economic Theory* **9**, 43–52.

Appendix

Proof. [Corollary 6.6.4] Let us consider any two objects $a, b \in A$, such that aDb, and let us denote by $D_2^{'}$ the dominance relation over set $A \times A$, defined in the same way as the dominance relation D_2 over set B, with the only difference that B (appearing in the definition of D_2) is replaced by $A \times A$. First, let us observe that aDb implies that $(a, b)D_2^{'}(b, a)$ and, moreover, given any object $c \in A\backslash\{a, b\}$, it is true that $(a, c)D_2^{'}(b, c)$ and $(c, b)D_2^{'}(c, a)$. Secondly, note that every decision rule $r_S \in R_S$ that covers the dominated (w.r.t. $D_2^{'}$) pair of objects (b, a) [respectively, (b, c), (c, a)], covers also the dominating (w.r.t. $D_2^{'}$) pair of objects (a, b) [respectively, (a, c), (c, b)]. Analogously, every decision rule $r_{S^C} \in R_{S^C}$ that covers the dominating pair of objects (a, b) [respectively, (a, c), (c, b)], covers also the dominated pair of objects (b, a) [respectively, (b, c), (c, a)]. Therefore, after application of decision rules on set A, according to definitions (6.18) and (6.19) we get:

- $\mathcal{S}(a, b) \geq \mathcal{S}(b, a)$ and $\mathcal{S}^{\mathcal{C}}(b, a) \geq \mathcal{S}^{\mathcal{C}}(a, b)$;

- $\mathcal{S}(a,c) \geq \mathcal{S}(b,c)$ and $\mathcal{S}^{\mathcal{C}}(c,a) \geq \mathcal{S}^{\mathcal{C}}(c,b)$;
- $\mathcal{S}(c,a) \leq \mathcal{S}(c,b)$ and $\mathcal{S}^{\mathcal{C}}(a,c) \leq \mathcal{S}^{\mathcal{C}}(b,c)$.

Thus, from (6.21), we get $\tilde{\mathcal{R}}(a,b) \geq \tilde{\mathcal{R}}(b,a)$ and, moreover, $\tilde{\mathcal{R}}(a,c) \geq \tilde{\mathcal{R}}(b,c)$, and $\tilde{\mathcal{R}}(c,a) \leq \tilde{\mathcal{R}}(c,b)$. This set of inequalities is the antecedent of the implication given in Definition 6.6.3 of property CC. Thus, from this definition, we have $a \succeq (A, \tilde{\mathcal{R}})b$.

7

About the application of evidence theory in multicriteria decision aid

Mohamed Ayman Boujelben[1] and Yves De Smet[2]

[1]*GIAD, Faculté des Sciences Economiques et de Gestion de Sfax, Université de Sfax, Tunisia*
[2]*CoDE-SMG, Ecole polytechnique de Bruxelles, Université libre de Bruxelles, Belgium*

7.1 Introduction

Selecting an investment project (Zavadskas *et al*. 2008), ranking companies according to their performances (Augusto *et al*. 2008), identifying the disease of a patient (Belacel 2000), and assigning firms into homogeneous credit risk classes (Doumpos and Zopounidis 2002) are just a few examples of decision problems that involve the explicit consideration of multiple criteria. Multicriteria decision aid (MCDA; Roy and Bouyssou 1993; Vincke 1992) is a discipline that has been developed to help a decision maker in solving problems where several conflicting criteria should be considered. Within this field, authors generally distinguish three main problems: the choice, the ranking and the sorting. The first one consists of choosing a subset of actions considered as the most interesting ones. The second problem consists of ordering the actions from the best one to the worst one by building partial and/or total preorders. Finally, the third problem

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

refers to the assignment of actions into predefined classes (groups or categories). Several multicriteria methods have been proposed to deal with these problems. Generally, these approaches are divided into three major trends (Vincke 1992): multiattribute utility theory (MAUT; Fishburn 1970; Keeney and Raiffa 1976), outranking methods (ELECTRE approaches, Figueira *et al*. 2005b; PROMETHEE, Brans and Mareschal 2005; Brans and Vincke 1985) and interactive methods (Vincke 1992). In what follows, we will assume that the reader is familiar with the MCDA field. If this is not the case, we invite the reader to consult Figueira *et al*. (2005a) for recent state of the art surveys.

Preference modeling is a fundamental step in MCDA. It involves the identification of assessment grades, criteria weights, preference, concordance and discordance thresholds, etc. The elicitation of this kind of information by the decision maker remains a complex and delicate task. As a result, different forms of imperfection in data such as imprecision and uncertainty have to be managed. This can be due for instance to the subjectiveness of human behavior, to inexact measurements or to the unstable characteristic of some values (Roy 1989).

Several models have been used in MCDA to tackle imperfect data. For instance, probability theory (Williams 1991) has been applied in PROMETHEE to deal with stochastic evaluations and uncertain criteria weights (Hyde *et al*. 2003). Furthermore, fuzzy sets theory (Dubois and Prade 1980; Zadeh 1965) has been used in the analytic hierarchy process (AHP) to express fuzzy judgments between the pairs of actions on each criterion (Deng 1999). Other approaches such as possibility theory (Dubois and Prade 1988a; Zadeh 1978) and evidence theory (Shafer 1976) have also been applied in MCDA to represent different forms of imperfection in data. The use of evidence theory in the multicriteria analysis will be at the core of this chapter.

Evidence theory, also called Dempster–Shafer theory or belief functions theory, is a convenient framework for modeling imperfect information and for combining it. This formalism has been proposed as a generalization of subjective probability theory and a model that represents uncertainty and total ignorance. It has also been the starting point of many developments such as the transferable belief model (Smets 1988; Smets and Kennes 1994). Moreover, it has well understood connections to other frameworks such as probability and possibility theories.

Evidence theory has primarily been developed in the context of artificial intelligence. For instance, one may cite pattern classification (Denoeux and Smets 2006; Xu *et al*. 1992), and clustering (Denoeux and Masson 2004; Masson and Denoeux 2008). In the field of MCDA, it has been used, to the best of our knowledge, in five multicriteria methods. The first approach is a ranking procedure called the evidential reasoning algorithm (Yang 2001; Yang and Singh 1994; Yang and Xu 2002) which allows to deal with problems where the evaluations of the actions are represented by belief structures. The second method is an extension of AHP modeled by evidence theory and called the Dempster–Shafer/analytic hierarchy process (DS/AHP) (Beynon *et al*. 2000). The third approach is a sorting method called DISSET based on evidence theory (Boujelben *et al*. 2007) where the information about the categories is represented by reference alternatives with precise and/or imprecise labels. The fourth model is a choice method inspired by ELECTRE I where the evaluations of the actions and the criteria weights are expressed by belief distributions (Boujelben *et al*. 2009a). Finally, the fifth procedure is a ranking model inspired by Xu *et al*.'s method where the evaluations are also represented by belief structures (Boujelben *et al*. 2011).

The objective of this chapter is to summarize these methods and to illustrate how evidence theory has been applied in the field of MCDA. It is organized as follows: in Section 7.2 we introduce the basic notions of evidence theory, then two new concepts developed within this framework for MCDA are presented in Section 7.3 and the five methods cited above are described in Section 7.4. Finally, a discussion on the application of evidence theory in the field of MCDA is proposed in Section 7.5.

7.2 Evidence theory: Some concepts

Evidence theory was initially developed by Arthur Dempster in 1967 (Dempster 1967), formalized by Glenn Shafer in 1976 (Shafer 1976) and axiomatically justified by Philippe Smets in his transferable belief model (Smets 1988; Smets and Kennes 1994). Generally, this approach is represented by three fundamental steps: knowledge modeling, combination and decision making. In this section, the main concepts of this theory are briefly introduced. The interested reader can refer to Yager and Liu (2008) for further details and recent state of the art surveys.

7.2.1 Knowledge model

Basically, imperfection in data within evidence theory is modeled using a function called the basic belief assignment (BBA). This function is defined on a finite set of mutually exclusive and exhaustive statements called the frame of discernment. Let $\Theta = \{S_1, S_2, \ldots, S_n\}$ be this set and 2^Θ be the powerset of Θ. A BBA (Shafer 1976) is a function m defined from 2^Θ to [0, 1] such as $m\{\emptyset\} = 0$ and $\sum_{A\subseteq\Theta} m(A) = 1$. The quantity $m(A)$, called the basic belief mass of subset A, represents the partial belief that A is true, i.e., the belief committed exactly to A. When $m(A) \neq 0$, A is called a focal element or a focal set.

The concept of BBA allows representing every state of beliefs even the total ignorance case. This state (which cannot be modeled in probability theory) is easily represented in evidence theory using a particular form of BBA called vacuous BBA. This type is defined such as the total belief is assigned to Θ, i.e., $m(\Theta) = 1$ and $m(A) = 0$ for all $A \subset \Theta$. Furthermore, as already mentioned, evidence theory has been proposed as an approach that generalizes subjective probability theory. Indeed, a BBA corresponds naturally to a probability function when all its focal elements are singletons. This particular case of BBA is referred to as Bayesian BBA.

A BBA can equivalently be represented by its associated belief (or credibility) and plausibility functions (Shafer 1976) defined from 2^Θ to [0, 1], respectively, as follows:

$$Bel(A) = \sum_{\substack{B\subseteq A \\ B\neq\emptyset}} m(B) \tag{7.1}$$

$$Pl(A) = \sum_{A\cap B\neq\emptyset} m(B) \tag{7.2}$$

The quantity $Bel(A)$ measures the total belief that supports completely A whereas the plausibility $Pl(A)$ quantifies the total belief that can potentially be placed in A,

i.e., the belief that supports completely and partially A. Of course, $Bel(\emptyset) = Pl(\emptyset) = 0$, $Bel(\Theta) = Pl(\Theta) = 1$ and $Bel(A) \leq Pl(A)$ for all $A \subseteq \Theta$. Moreover, these two functions are naturally connected by the relation $Pl(A) = 1 - Bel(\bar{A})$ where $\bar{A}$ is the complement of A for all $A \subseteq \Theta$.

Finally, let us note that when a BBA m is provided by a not fully reliable source, this unreliability is taken into account within evidence theory through the discounting operation. This is performed using the concept of discounting factor $\alpha \in [0, 1]$ which quantifies the reliability of the source: the closer to 1, the greater the reliability. Formally, the discounted BBA denoted m^α is defined as follows:

$$
\begin{aligned}
m^\alpha(A) &= \alpha.m(A) \text{ for all } A \subset \Theta \\
m^\alpha(\Theta) &= m(\Theta) + (1 - \alpha). \sum_{A \subset \Theta} m(A)
\end{aligned}
\tag{7.3}
$$

Of course, discounting with a factor 0 leads to a vacuous BBA whereas a factor 1 keeps the BBA unchanged.

7.2.2 Combination

The combination is an operation that plays a central role in evidence theory. Indeed, given several BBAs induced by different sources, it is usually required to aggregate them in order to yield a global BBA synthesizing their knowledge. Within this context, several combination rules have been proposed (Smets 2007).

Dempster's rule of combination (Shafer 1976), also called the normalized conjunctive rule, is the most commonly used operator in the combination of BBAs provided by independent sources, i.e., distinct BBAs (Smets 2007). This rule is defined in the case of two sources as follows:

$$
m_{1|2}(A) = (1 - k)^{-1}. \sum_{B \cap C = A} m_1(B).m_2(C)
\tag{7.4}
$$

where m_1 and m_2 are the two distinct BBAs to be combined, $m_{1|2} = m_1 \oplus m_2$ is the BBA resulting from their combination and $k = \sum_{B \cap C = \emptyset} m_1(B).m_2(C)$ is the belief mass that the combination assigns to the empty set. The coefficient k reflects the conflict between the sources whereas the quotient $(1 - k)^{-1}$ is a normalization factor guaranteeing that no belief is associated with the empty set and that the total belief remains equal to one.

Dempster's rule has several interesting mathematical properties. It can be proved to be both commutative and associative. Therefore, the combination result of several BBAs is independent of the order in which they are considered. This rule has extensively been used in expert systems (Beynon *et al*. 2001; Biswas *et al*. 1988). However, it can lead in some situations to unsatisfactory results since it can assign the total mass to a minority opinion and it can lead to a loss of the majority opinion (Yamada 2008). For that purpose, several combination rules have been proposed. Among others, we can mention Yager's rule (Yager 1987), Dubois and Prade's rule (Dubois and Prade 1988b), Murphy's rule (Murphy 2000), and Yamada's rule (Yamada 2008). Moreover, Dempster's rule cannot be applied to combine BBAs given by dependent sources, i.e., nondistinct BBAs. A recent combination operator called the cautious rule (Denoeux 2008) has been proposed to perform combination in such a situation. The interested reader can refer to Ha-Duong (2008) where an application of this rule to climate sensitivity assessment is given.

7.2.3 Decision making

As the combination, the decision making constitutes a crucial component of evidence theory. Several decision rules based on the couple ‘belief, plausibility’ have been developed in this context. Among others, we can mention the maximum of belief rule (Janez 1996) which consists of choosing the most credible statement(s), and the maximum of plausibility rule (Appriou 1991) which allows the selection of the most plausible one(s). Other rules have also been proposed which are based on techniques that transform a BBA into a function having similar properties than a probability function. For instance, the pignistic transformation (Smets 1990, 2002, 2005) allows the transformation of a BBA into a pignistic probability function. This procedure consists of distributing equally each belief mass $m(A)$ among the statements that compose A. Formally, the pignistic probability function $BetP$ is defined as follows:

$$BetP\,(S_i) = \sum_{\substack{A \subseteq \Theta \\ S_i \in A}} \frac{1}{|A|} . \frac{m(A)}{1 - m(\emptyset)} \tag{7.5}$$

where $|A|$ is the cardinality of A. In this context, the decision consists of choosing the statement(s) having the maximum of pignistic probability.

7.3 New concepts in evidence theory for MCDA

A criterion is defined as a mapping of the alternatives into a totally ordered set (Vincke 1992). As a consequence, the evaluation of the actions according to a given criterion can always be ranked from the worst one to the best one (with eventually some ties). The relative importance of criteria weights generally allows to order them. In sorting[1] problems, categories are defined by their ranks. Order relation plays therefore a central role in the field of MCDA. Of course, the concept of BBA can be used to represent imperfect data in this context. Nevertheless, according to us, it has to take into account order relations which are omnipresent in MCDA. In this section, we will describe two recent concepts in evidence theory applied to MCDA. The first one is an extension of the dominance principle while the second one allows the distance between two BBAs to be quantified.

7.3.1 First belief dominance

The first belief dominance (Boujelben *et al*. 2009a,b) is a generalization of the first stochastic dominance (Hadar and Russell 1969) that allows pairwise comparisons between BBAs defined on a frame consisting of ordered elements. Let $\Theta = \{x_1, x_2, \ldots, x_r\}$ be this set such as $x_1 \prec x_2 \prec \ldots \prec x_r$ (where $\prec$ denotes ‘less preferred than’) and let m_i and m_j be two BBAs defined on Θ. In such a case, since the elements of Θ are ordered from the worst one to the best one, the focal elements should be either singletons or disjunctions of successive elements of Θ (sets such as $\{x_1, x_3\}$ or $\{x_1, x_2, x_4\}$ are not meaningful since the elements composing them are not successive). In what follows, we will denote the

[1] In MCDA, the classification problem covers two distinct situations: the sorting (or the ordinal classification) if the classes are ordered from the worst one to the best one and the nominal classification if they are defined in a nominal way (Zopounidis and Doumpos 2002).

set of these focal elements by $S(\Theta)$. Moreover, for all $k \in \{0, \ldots, r\}$, let:

$$A_k = \begin{cases} \emptyset & \text{if} \quad k = 0 \\ \{x_1, \ldots, x_k\} & \text{otherwise} \end{cases} \tag{7.6}$$

and let $\overrightarrow{S}(X)$ denote the set $\{A_1, A_2, \ldots, A_r\}$. Similarly, for all $l \in \{0, \ldots, r\}$ such as $l = r - k$, let:

$$B_l = \begin{cases} \emptyset & \text{if} \quad l = 0 \\ \{x_{r-l+1}, \ldots, x_r\} & \text{otherwise} \end{cases} \tag{7.7}$$

and let $\overleftarrow{S}(X)$ denote the set $\{B_1, B_2, \ldots, B_r\}$. k and l represent, respectively, the number of elements constituting the sets A_k and B_l. Of course, $|\overrightarrow{S}(X)| = |\overleftarrow{S}(X)| = r$, $\bar{A_k} = B_{r-k} = B_l$ for all $k \in \{0, \ldots, r\}$ and $\bar{B_l} = A_{r-l} = A_k$ for all $l \in \{0, \ldots, r\}$.

The first belief dominance concept is based on the notions of the ascending and descending belief functions (Boujelben *et al*. 2009a,b) which are defined as follows:

Definition 7.3.1 *The ascending belief function induced by* m_i *is a function* $\overrightarrow{Bel_i} : \overrightarrow{S}(X) \to [0, 1]$ *defined such as* $\overrightarrow{Bel_i}(A_k) = \sum_{C \subseteq A_k} m_i(C)$ *for all* $A_k \in \overrightarrow{S}(X)$.

Definition 7.3.2 *The descending belief function induced by* m_i *is a function* $\overleftarrow{Bel_i} : \overleftarrow{S}(X) \to [0, 1]$ *defined such as* $\overleftarrow{Bel_i}(B_l) = \sum_{C \subseteq B_l} m_i(C)$ *for all* $B_l \in \overleftarrow{S}(X)$.

These two functions allow taking into account implicitly the fact that $x_1 \prec x_2 \prec \ldots \prec x_r$. Indeed, $\overrightarrow{Bel_i}$ represents the beliefs of the nested sets $A_1, A_2, \ldots$ and A_r, i.e., the sets $\{x_1\}, \{x_1, x_2\}, \ldots$ and $\{x_1, \ldots, x_r\}$ whereas $\overleftarrow{Bel_i}$ represents the beliefs of the nested sets $B_1, B_2, \ldots$ and B_r, i.e., the sets $\{x_r\}, \{x_{r-1}, x_r\}, \ldots$ and $\{x_1, \ldots, x_r\}$. Of course, since x_1 and x_r are, respectively, the worst and the best elements of Θ, the more the values of $\overrightarrow{Bel_i}$ decrease and those of $\overleftarrow{Bel_i}$ increase, the better is the BBA. As a result, the first belief dominance is defined as follows:

Definition 7.3.3 m_i *is said to dominate* m_j *according to the first belief dominance if and only if the following two conditions are verified simultaneously:* $\overrightarrow{Bel_i}(A_k) \leq \overrightarrow{Bel_j}(A_k)$ *for all* $A_k \in \overrightarrow{S}(X)$ *and* $\overleftarrow{Bel_i}(B_l) \geq \overleftarrow{Bel_j}(B_l)$ *for all* $B_l \in \overleftarrow{S}(X)$.

In the case where the two conditions are not verified simultaneously, m_i does not dominate m_j according to the first belief dominance concept. Therefore, two situations can be identified when using this approach:

- The dominance, if m_i dominates m_j (m_i FBD m_j);
- The nondominance, if m_i does not dominate m_j (m_i $\overline{FBD}$ m_j).

Finally, it is worth mentioning that similar approaches to the first belief dominance approach called credal orderings have been proposed by Thierry Denoeux (Denoeux 2009).

7.3.2 RBBD concept

Recently, a new concept in evidence theory called RBBD (**r**anking **B**BAs based on **b**elief **d**istances) has been proposed in Boujelben *et al*. (2011). This approach allows ranking BBAs defined on a frame consisting of ordered elements. The underlying idea of this concept has been inspired from a ranking method called TOPSIS (Hwang and Yoon 1981) which is based on the comparison of the actions to ideal and nadir actions. In RBBD, the BBAs to be ranked are compared with the ideal BBA and nadir BBA. The former is the best BBA among all the BBAs that can be defined on the set $S(\Theta)$ whereas the latter is the worst one among these BBAs. Therefore, since x_1 and x_r are, respectively, the worst and the best subsets of $S(\Theta)$, the ideal and nadir BBAs are, respectively, the BBAs $m_{ideal}(\{x_r\}) = 1$ and $m_{nadir}(\{x_1\}) = 1$.

The comparison of the BBAs with the ideal or nadir BBA is based on the notions of ascending and descending belief distances proposed in Boujelben *et al*. (2011).

Definition 7.3.4 *Let $\overrightarrow{Bel_i}$ and $\overrightarrow{Bel_j}$ be two ascending belief functions related, respectively, to two BBAs m_i and m_j, the ascending belief distance is defined as follows:* $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_j}\right) = \sum_{k=1}^{r} \mid \overrightarrow{Bel_i}(A_k) - \overrightarrow{Bel_j}(A_k) \mid$.

Definition 7.3.5 *Let $\overleftarrow{Bel_i}$ and $\overleftarrow{Bel_j}$ be two descending belief functions related, respectively, to two BBAs m_i and m_j, the descending belief distance is defined as follows:* $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_j}\right) = \sum_{l=1}^{r} \mid \overleftarrow{Bel_i}(B_l) - \overleftarrow{Bel_j}(B_l) \mid$.

The dissimilarity of a given BBA m_i to the ideal BBA m_{ideal} is measured through the two ascending and descending belief distances $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right)$. Since m_{ideal} is a particular type of BBA, these belief distances are, respectively, expressed as follows:

$$d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right) = \left(\sum_{k=1}^{r} \overrightarrow{Bel_i}(A_k)\right) - 1 \tag{7.8}$$

$$d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right) = r - \sum_{l=1}^{r} \overleftarrow{Bel_i}(B_l) \tag{7.9}$$

Similarly, the comparison of m_i to the nadir BBA m_{nadir} is performed using the two ascending and descending belief distances $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right)$. Since m_{nadir} is a particular case of BBA, these belief distances can, respectively, be written as:

$$d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right) = r - \sum_{k=1}^{r} \overrightarrow{Bel_i}(A_k) \tag{7.10}$$

$$d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right) = \left(\sum_{l=1}^{r} \overleftarrow{Bel_i}(B_l)\right) - 1 \tag{7.11}$$

Based on the belief distances between the BBAs and the ideal BBA, it is possible to obtain two rankings of the BBAs. The first one is associated with the ascending belief distances $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right)$ whereas the second one is related to the descending belief

distances $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right)$. Of course, the lower the values of these belief distances, the better m_i. Similarly, the comparison of the BBAs with the nadir BBA allows two rankings to be obtained related, respectively, to the ascending and descending belief distances $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right)$: the higher their values, the better m_i.

From Equation (7.8), Equation (7.9), Equation (7.10) and Equation (7.11), it is easy to see that the ascending belief distances to the ideal and nadir BBAs are linearly linked (it is also the case for the descending belief distances to the ideal and nadir BBAs). Since the belief distances to the ideal BBA are to be minimized and those to the nadir BBA are to be maximized, one can deduce that the rankings related to $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right)$ are the same [similarly for the rankings related to $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right)$]. Therefore, the comparison with the ideal and nadir BBAs can be reduced to a comparison with one of them.

When comparing the BBAs with the ideal BBA, it is possible to deduce a partial preorder of these BBAs called the RBBD I ranking. This preorder is obtained as the intersection of the two rankings induced by $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right)$. Three preference situations can be distinguished in this context between the BBAs: the preference (P), the indifference (I) and the incomparability (J). The latter appears between two BBAs when it is impossible to express indifference or preference between them. Formally, these relations can be expressed as follows:

- $m_i \; P \; m_j \Leftrightarrow$
  $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right) \leq d\left(\overrightarrow{Bel_j}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right) < d\left(\overleftarrow{Bel_j}, \overleftarrow{Bel_{ideal}}\right)$
  $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right) < d\left(\overrightarrow{Bel_j}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right) \leq d\left(\overleftarrow{Bel_j}, \overleftarrow{Bel_{ideal}}\right)$
- $m_i \; I \; m_j \Leftrightarrow$
  $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right) = d\left(\overrightarrow{Bel_j}, \overrightarrow{Bel_{ideal}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right) = d\left(\overleftarrow{Bel_j}, \overleftarrow{Bel_{ideal}}\right)$
- $m_i \; J \; m_j \Leftrightarrow$ Otherwise

Using the belief distances to the nadir BBA, the RBBD I ranking can be obtained as the intersection of the two rankings induced by $d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right)$ and $d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right)$.

In addition to the partial preorder, it is possible to obtain a total preorder of the BBAs, i.e., a complete ranking without incomparabilities. This preorder, called the RBBD II ranking, can be deduced using the belief distances to the ideal BBA or to the nadir BBA. When the BBAs are compared with the ideal BBA, the total preorder is established based on a global score α_i that can be computed for each BBA m_i as follows:

$$\alpha_i = d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{ideal}}\right) + d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{ideal}}\right) \tag{7.12}$$

Of course, the lower α_i, the better m_i. Using the belief distances to the nadir BBA, the RBBD II ranking is deduced using a global score β_i that can be defined for each BBA m_i as follows:

$$\beta_i = d\left(\overrightarrow{Bel_i}, \overrightarrow{Bel_{nadir}}\right) + d\left(\overleftarrow{Bel_i}, \overleftarrow{Bel_{nadir}}\right) \tag{7.13}$$

Obviously, the higher β_i, the better m_i. Moreover, it is easy to deduce from Equation (7.8), Equation (7.9), Equation (7.10), and Equation (7.11) that $\alpha_i = -\beta_i - 2 + 2.r$. Therefore, α_i and β_i are linearly linked.

Before ending this section, let us note that the RBBD concept is coherent with the first belief dominance approach (Boujelben *et al*. 2011). Indeed, if m_i FBD m_j, the rank of m_i may not be worse than the rank of m_j using the RBBD concept. Similarly, let us note that it is possible to use the first belief dominance to obtain a ranking of the BBAs. However, the main drawback of using this approach to rank the BBAs is the number of incomparabilities between the BBAs that can be important is some situations. This number is always superior or equal to the one obtained by the RBBD concept (the RBBD I ranking; Boujelben *et al*. 2011).

7.4 Multicriteria methods modeled by evidence theory

This section is devoted to a review of five multicriteria approaches modeled by evidence theory mentioned in Section 7.1. Initially, this theory has been used for modeling the evidential reasoning algorithm and the DS/AHP method. Then, the three other approaches (i.e., DISSET, the choice method inspired by ELECTRE I and the ranking model inspired by Xu *et al*.'s method) have been proposed in Boujelben (2011) where the objective has been to apply evidence theory in the three main multicriteria problematics. In what follows, we will denote $\mathcal{A} = \left\{a_1, a_2, \ldots, a_n\right\}$ the set of actions and $G = \left\{g_1, g_2, \ldots, g_q\right\}$ the set of criteria. Other specific notations related to each method will be defined when needed.

7.4.1 Evidential reasoning approach

The evidential reasoning algorithm (Yang 2001; Yang and Singh 1994; Yang and Xu 2002) is an approach designed to manage ranking problems where the actions are evaluated on a set of ordinal criteria and where the evaluations are given imperfectly. In what follows, let $X = \left\{x_1, x_2, \ldots, x_r\right\}$ be the set of assessment grades used to evaluate the actions on all the criteria. We assume that $x_1 \prec x_2 \prec \ldots \prec x_r$. Moreover, we suppose that a utility function denoted u is defined on this set. Of course, $u(x_1) < u(x_2) < \ldots < u(x_r)$.

The evaluation of an action a_i on a criterion g_h is described by a belief structure β_i^h defined on X that takes into account uncertainty and incompleteness when evaluating a_i. $\beta_i^h(x_j)$ (with $j \in \{1, \ldots, r\}$) denotes the belief degree of assessing a_i on g_h according to assessment grade x_j. This degree is defined such as $\beta_i^h(x_j) \geq 0$. The assessment of a_i on g_h is said to be complete when $\sum_{j=1}^{r} \beta_i^h(x_j) = 1$ and incomplete when $\sum_{j=1}^{r} \beta_i^h(x_j) < 1$.[2]

In order to rank the actions, the evidential reasoning algorithm proceeds as follows. At first, each belief structure β_i^h is transformed into a BBA denoted m_i^h. This is performed by multiplying β_i^h by the criterion weight w_h (where $w_h > 0$ and $\sum_{h=1}^{q} w_h = 1$). This

[2] In this approach, the belief assessments are limited to single evaluation grades (not on subsets as in Sections 7.4.4 and Section 7.4.5).

leads to the following BBA:

$$\begin{aligned}
m_i^h(x_1) &= w_h.\beta_i^h(x_1)\\
&\vdots\\
m_i^h(x_r) &= w_h.\beta_i^h(x_r)\\
m_i^h(X) &= 1-\sum_{j=1}^{r} m_i^h(x_j) = 1 - w_h.\sum_{j=1}^{r}\beta_i^h(x_j)
\end{aligned}\tag{7.14}$$

At this point, let us note that it is important to distinguish the two belief masses $\overline{m}_i^h(X)$ and $\hat{m}_i^h(X)$ that compose $m_i^h(X)$. Formally, $m_i^h(X) = \overline{m}_i^h(X) + \hat{m}_i^h(X)$where:

$$\overline{m}_i^h(X) = 1 - w_h \tag{7.15}$$

$$\hat{m}_i^h(X) = w_h.\left(1-\sum_{j=1}^{r}\beta_i^h(x_j)\right) \tag{7.16}$$

Indeed, $\overline{m}_i^h(X)$ is the belief mass that appears when multiplying β_i^h by the criterion weight w_h. Since this mass is not assigned to any set, it is transferred to X. On the other hand, $\hat{m}_i^h(X)$ is the belief mass which is caused by the incompleteness in the assessment of a_i on g_h. The utility of this decomposition will appear in the next step when combining the BBAs.

Once all the BBAs characterizing the evaluations of a_i on all the criteria are determined, Dempster's rule is applied to aggregate them as follows:

$$m_i = m_i^1 \oplus m_i^2 \oplus \ldots \oplus m_i^q \tag{7.17}$$

Of course, since the focal elements of m_i^h belong to the set $\{\{x_1\}, \ldots, \{x_r\}, X\}$ and since Dempster's rule is a conjunctive operator, the focal elements of m_i belong also to $\{\{x_1\}, \ldots, \{x_r\}, X\}$. Moreover, it is important to decompose the combined belief mass $m_i(X)$ into two belief masses $\overline{m}_i(X)$ and $\hat{m}_i(X)$. The former represents the combined belief mass caused by taking into account the criteria weights and is given by:

$$\overline{m}_i(X) = \prod_{h=1}^{q}\overline{m}_i^h(X) = \prod_{h=1}^{q}(1-w_h) \tag{7.18}$$

The latter reflects the combined belief mass representing the incompleteness in the global evaluation of a_i and is defined as follows:

$$\overline{m}_i(X) = m_i(X) - \overline{m}_i(X) \tag{7.19}$$

Therefore, in order to obtain a combined belief structure that represents the global evaluation of a_i, the authors propose to normalize m_i by dividing it by $1 - \overline{m}_i(X)$. This is due to the fact that $\overline{m}_i(X)$ does not reflect the incompleteness in the assessment of a_i. This leads to the following combined belief structure:

$$\beta_i(x_1) = \frac{m_i(x_1)}{1-\overline{m}_i(X)}$$

$$\vdots$$

$$\beta_i(x_r) = \frac{m_i(x_r)}{1 - \overline{m}_i(X)}$$

$$\beta_i(X) = \frac{\hat{m}_i(X)}{1 - \overline{m}_i(X)} \tag{7.20}$$

Of course, the global assessment of a_i is complete when $\sum_{j=1}^{r} \beta_i(x_j) = 1$ (i.e., when $\beta_i(X) = 0$) and incomplete when $\sum_{j=1}^{r} \beta_i(x_j) < 1$ (i.e., when $\beta_i(X) > 0$). Let us note that, in the original version of the evidential reasoning approach (Yang and Singh 1994), $\overline{m}_i(X)$ and $\hat{m}_i(X)$ were not defined and therefore the normalization was not required to determine the combined belief structure. Indeed, it is given directly by:

$$\beta_i(x_1) = m_i(x_1)$$

$$\vdots$$

$$\beta_i(x_r) = m_i(x_r)$$

$$\beta_i(X) = m_i(X) \tag{7.21}$$

This approach has then been revised as described above. The interested reader can refer to Yang (2001) and Yang and Xu (2002) for further details.

Then, based on the combined belief structure and the assessment grades utilities $u(x_1)$, $u(x_2)$, $\ldots$ and $u(x_r)$, an expected utility denoted $u(a_i)$ is computed when the global assessment of a_i is complete. This is performed using the following relation:

$$u(a_i) = \sum_{j=1}^{r} \beta_i(x_j).u(x_j) \tag{7.22}$$

When the global assessment of a_i is incomplete, an expected utility interval denoted $[u_{\min}(a_i), u_{\max}(a_i)]$ is computed as follows:

$$u_{min}(a_i) = (\beta_i(x_1) + \beta_i(X)).u(x_1) + \sum_{j=2}^{r} \beta_i(x_j).u(x_j) \tag{7.23}$$

$$u_{max}(a_i) = \sum_{j=1}^{r-1} \beta_i(x_j).u(x_j) + (\beta_i(x_r) + \beta_i(X)).u(x_r) \tag{7.24}$$

An average expected utility can also be computed using the following relation:

$$u_{avg}(a_i) = \frac{u_{min}(a_i) + u_{max}(a_i)}{2} \tag{7.25}$$

Finally, the actions are ranked based on their expected utility intervals. An action a_i is said to be preferred to another action $a_{i'}$ if $u_{min}(a_i) > u_{max}(a_{i'})$. They are indifferent if $u_{min}(a_i) = u_{min}(a_{i'})$ and $u_{max}(a_i) = u_{max}(a_{i'})$. Otherwise, the average expected utility can be used to rank the actions.

The evidential reasoning approach has been applied in several decision problems for instance in engineering design (Yang and Sen 1997), contractor selection (Sönmez *et al*. 2001), and safety analysis (Wang and Yang 2001). In addition, it has been extended to handle both qualitative and quantitative criteria in uncertain context using transformation techniques (Yang 2001). This approach can also be used in multicriteria problems where the evaluations are modeled by interval belief degrees (Wang *et al*. 2006) or are determined on the basis of fuzzy assessment grades (Yang *et al*. 2006). In the same way, it can be applied to deal with both interval belief degrees and interval weights using nonlinear optimization models (Guo *et al.* 2007).

However, the main drawback of this method is that the belief structures expressing the performances of the actions are defined using the same set of assessment grades on all criteria. In a multicriteria context, the criteria can be defined such as their related assessment grades sets are different. Moreover, the belief structure β_i^h expressing the assessment of a_i on g_h takes into account uncertainty and incompleteness when evaluating a_i but does not consider imprecision. For instance, the decision maker can hesitate between two assessment grades x_1 and x_2 when evaluating a_i but without being able to assign a belief degree to each of them. Within evidence theory, this can be modeled by a belief structure where the set $\{x_1, x_2\}$ is a focal element, i.e., where a belief degree is assigned to this set. This aspect will be further developed in Section 7.4.4 and Section 7.4.5.

7.4.2 DS/AHP

DS/AHP (Beynon *et al.* 2000) is a ranking method which incorporates evidence theory with the philosophy of AHP (Saaty 1980). This approach has been proposed to avoid some difficulties of AHP. Indeed, one limit of this method is the high number of judgments that the decision maker has to give on the actions set. This number rises quickly as the numbers of alternatives and criteria rise and is equal to $q.n.(n-1)/2$ judgments (where n is the number of actions and q is the number of criteria). Moreover, another difficulty of AHP is the problem of possible inconsistency related to the judgments.

In order to rank the actions, the DS/AHP method proceeds as follows. At the first step, the decision maker has to give, for each criterion, his judgments not on all the pairs of alternatives as in AHP, but on groups of actions chosen by him and globally compared with the set $\mathcal{A}$. These judgments express the importance of these groups with regard to $\mathcal{A}$ on each criterion g_h and are evaluated on the basis of a judgment scale chosen by the decision maker. This leads to the definition of a comparison matrix denoted B_h on each criterion g_h where the rows and the columns are the set $\mathcal{A}$ and the groups of actions compared with it. Table 7.1 gives an example of a comparison matrix where two groups of actions $\{a_1\}$ and $\{a_2, a_3\}$ are compared with the set $\mathcal{A} = \{a_1, a_2, a_3\}$ and where the importance degrees of these groups with regard to $\mathcal{A}$ is equal to 6 and 4. Of course, since the groups of actions are not compared with each other, the importance degree between them is equal to 0. Moreover, let us note the 1 values that appear in the diagonal of this matrix indicate that two equivalent sets of actions have obviously the same importance.

The groups of actions compared with $\mathcal{A}$ are defined such as there is no intersection between them, i.e., no action(s) can be considered simultaneously in two groups. This obviously allows the problem of inconsistency of the judgments to be avoided since these groups are not compared with each other but to the set $\mathcal{A}$. Moreover, the comparison

Table 7.1 Comparison matrix.

	$\{a_1\}$	$\{a_2, a_3\}$	$\mathcal{A}$
$\{a_1\}$	1	0	6
$\{a_2, a_3\}$	0	1	4
$\mathcal{A}$	1/6	1/4	1

with $\mathcal{A}$ allows the number of judgments to be reduced considerably which is equal, in the worst case, to $q.n$ (n is the maximal number of subsets that can be defined on $\mathcal{A}$ such as there is no intersection between them).

In the second step, the criteria weights are determined as in the traditional AHP method. They are then incorporated in their respective comparison matrices. This is performed by multiplying the elements of the last column of B_h (except its last entry) by the respective criterion weight w_h. If α is the importance degree of a group of actions with regard to $\mathcal{A}$ on g_h, the resulting value issued from this multiplication is $\alpha.w_h$ [the resulting value in the bottom row of the matrix is $1/(\alpha.w_h)$]. Let us note that the values 0 and 1 of B_h are not affected by the multiplication by w_h since they do not reflect the judgments of the decision maker. In what follows, the matrix deduced from the incorporation of w_h in B_h will be denoted B'_h.

In the third step, the priority values of $\mathcal{A}$ and the groups of actions compared with it are determined on each criterion g_h as in AHP using the right eigenvector method (Saaty 1980). This is performed by normalizing the elements of the right eigenvector R_h related to the matrix B'_h and associated with the highest eigenvalue λ_h^{max}, i.e., $B'_h.R_h = \lambda_h^{max}.R_h$. These normalized values of R_h constitute a BBA denoted m_h defined on each criterion g_h where the focal elements are $\mathcal{A}$ and the groups of actions compared with $\mathcal{A}$ and where the belief masses are the priority values related to these sets.

Once all the BBAs m_h (with $h \in \{1, \ldots, q\}$) are determined, Dempster's rule of combination is then used to aggregate them. The resulting BBA, denoted m, gives the global priority values for $\mathcal{A}$ and groups of actions. Formally, m is defined as follows:

$$m = m_1 \oplus m_2 \oplus \ldots \oplus m_q \tag{7.26}$$

Finally, the actions are ranked iteratively starting by determining the best action(s). This is performed using the belief function *Bel* related to m. The steps of the ranking procedure can be explained as follows:

- Initially, we compute the degrees $Bel(\{a_i\})$ for all $a_i \in \mathcal{A}$. The action(s) that has (have) the maximum of belief is (are) ranked at the first position. Let us denote this set by D_1;
- Then, we compute the degrees $Bel(\{D_1, a_i\})$ for all $a_i \in \mathcal{A}\backslash\{D_1\}$. The action(s) that has (have) the maximum of belief is (are) ranked at the second position. Let us denote this set by D_2;
- ...
- We compute the degrees $Bel(\{D_1, D_2, \ldots, D_l, a_i\})$ for all $a_i \in \mathcal{A}\backslash\{D_1, D_2, \ldots, D_l\}$. The action(s) that has (have) the maximum of belief is (are) ranked at the $(l+1)^{th}$ position. Let us denote this set by D_{l+1};

- ...
- This procedure finishes when all the actions are ranked.

Let us note that it is also possible to rank the actions on the basis of the plausibility function Pl related to m. This is performed using the same mechanism described above, but instead of computing belief degrees, we compute plausibility degrees. The two rankings based on Bel and Pl are not of course necessarily the same. Similarly, let us mention that a mathematical analysis including an understanding of uncertainty and ignorance of the judgments within DS/AHP have been discussed in Beynon (2002, 2005b), and a software for this approach has been developed (Beynon *et al.* 2001).

The DS/AHP method has also been extended in the group decision making context (Beynon 2005a) where the members are nonequivalent in their importance within the group. Briefly, this extension proceeds in four steps. At first, the members have to compare groups of actions with $\mathcal{A}$ and then to determine the priority values of $\mathcal{A}$ and each group of actions as described previously. Let m_f be the individual BBA given by member M_f that expresses these priority values. In order to take into account the nonequivalence of importance of the group members, the individual BBAs given by them having less importance than the maximum of the importances are discounted and normalized. Let m_f^* be the induced BBA at this level. Then, Dempster's rule of combination is used to aggregate these BBAs in order to obtain a collective BBA that gives the global priority values:

$$m = m_1^* \oplus m_2^* \oplus \ldots \oplus m_s^* \tag{7.27}$$

The resulting BBA m is then exploited to obtain the collective preorder of the actions using the belief or plausibility ranking procedures described above.

Finally, let us note that although the major advantage of incorporating evidence theory into AHP is the reduction of the problem complexity, the performance of DS/AHP still needs to be evaluated considering complex hierarchical problems.

7.4.3 DISSET

DISSET (Boujelben *et al.* 2007) is an approach that deals with sorting problems where the information about the categories is represented by a learning set (LS), i.e., a set of alternatives and their related known labels. The distinctive feature of this method relies on the fact that both precise and imprecise labels can be handled. Indeed, information related to the LS is not always precise. One may for instance imagine real world problems where the decision maker is unable to clearly propose alternatives which are assigned to a unique category. For example, in credit risk contexts, a financial analyst may hesitate to assign a given company between two successive risk categories among several that are considered. Another source of imprecision relies on the fact that the LS may be the output of a multicriteria sorting method such as ELECTRE TRI (Roy and Bouyssou 1993; Yu 1992), where both pessimistic and optimistic assignments are proposed. If the parameters of the method are not available anymore, the only information related to the categories is summarized therefore to these imprecise assignments.

The label is precise when its related alternative belongs to a unique category. It is imprecise when it is represented by a disjunction of successive categories. In what follows, let $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ be the set of categories defined such as $C_1 \prec C_2 \prec \ldots \prec C_m$,

$X = \{x_1, x_2, \ldots, x_r\}$[3] be the LS and $\{C_{v_j}, \ldots, C_{w_j}\}$ be the label of $x_j \in X$ (where C_{v_j} is the lower category of x_j and C_{w_j} is the higher category of x_j). Of course, when $v_j = w_j$, the label of x_j is precise. Otherwise, it is imprecise.

In order to assign a_i to its appropriated category, DISSET proceeds in four steps: the comparison, the definition of the BBAs, the combination and the decision making (the assignment).

The first step consists comparing of a_i with each alternative x_j in order to obtain information about the possible assignments of a_i. The comparison is performed using the label of x_j and preference degrees between a_i and x_j. Let $\Pi(a_i, x_j)$ [$\Pi(x_j, a_i)$, respectively) be the preference degree for a_i (x_j, respectively] with regard to x_j (a_i, respectively) over all the criteria. These preference degrees, which can be obtained as in the PROMETHEE method (Brans and Vincke 1985), are defined such as $\Pi(a_i, x_j) \geq 0$, $\Pi(x_j, a_i) \geq 0$ and $0 < \Pi(a_i, x_j) + \Pi(x_j, a_i) \leq 1$. $\Pi(a_i, x_j)$ [$\Pi(x_j, a_i)$, respectively] quantifies the preference strength of a_i over x_j (x_j over a_i): the closer to 1, the higher the preference. On the other hand, the degree $1 - \Pi(a_i, x_j) - \Pi(x_j, a_i)$ quantifies the indifference between a_i and x_j. Therefore, when comparing a_i and x_j, we can deduce the following three situations:

- If a_i is preferred to x_j, then $a_i \in \{C_{v_j}, \ldots, C_m\}$.
- If x_j is preferred to a_i, then $a_i \in \{C_1, \ldots, C_{w_j}\}$.
- If a_i is indifferent to x_j, then $a_i \in \{C_{v_j}, \ldots, C_{w_j}\}$.

The second step of this method consists of defining the BBAs resulting from the comparison of a_i with the alternatives of the LS. Since a_i is compared with r alternatives, therefore r BBAs should be defined. The belief masses of each BBA are determined as follows. When a_i is preferred to x_j, the information deduced on a_i is converted into the following belief mass:

$$m_j\left(\left\{C_{v_j}, \ldots, C_m\right\}\right) = \Pi(a_i, x_j) \tag{7.28}$$

Similarly, when x_j is preferred to a_i, the information deduced on a_i is represented by the following belief mass:

$$m_j\left(\left\{C_1, \ldots, C_{w_j}\right\}\right) = \Pi(x_j, a_i) \tag{7.29}$$

In the same way, when a_i is indifferent to x_j, the information deduced on a_i is converted into the following belief mass:

$$m_j\left(\left\{C_{v_j}, \ldots, C_{w_j}\right\}\right) = 1 - \Pi(a_i, x_j) - \Pi(x_j, a_i) \tag{7.30}$$

[3] The notation $X = \{x_1, x_2, \ldots, x_r\}$ is used differently in this section. Indeed, whilst it represents the assessment grades in Section 7.4.1, it defines the LS in this section.

As a result, the comparison of a_i with x_j leads to the following BBA:

$$
\begin{aligned}
m_j\left(\left\{C_{v_j}, \ldots, C_m\right\}\right) &= \Pi(a_i, x_j)\\
m_j\left(\left\{C_1, \ldots, C_{w_j}\right\}\right) &= \Pi(x_j, a_i)\\
m_j\left(\left\{C_{v_j}, \ldots, C_{w_j}\right\}\right) &= 1 - \Pi(a_i, x_j) - \Pi(x_j, a_i)
\end{aligned} \tag{7.31}
$$

As one can notice, the focal elements of this BBA can be either unique categories or disjunctions of successive categories. In what follows, we will denote $S(\mathcal{C})$ the set defined by the categories and all the successive combinations of them.

In the third step of DISSET, the r BBAs deduced from the comparison of a_i with alternatives are aggregated using Dempster's rule in order to yield a global BBA m on which the decision will be made, i.e.:

$$m = m_1 \oplus m_2 \oplus \ldots \oplus m_r \tag{7.32}$$

Of course, since Dempster's rule of combination is a conjunctive operator and the focal elements of the BBAs are either unique categories or disjunctions of successive categories, the focal elements of the combined BBA should be single categories or sets of successive categories.

Finally, the combined BBA is exploited to assign action a_i to its appropriate category or categories. This is performed using the maximum of the belief mass rule which consists of choosing the category or the set of successive categories having the maximum of belief mass. More formally:

$$A^* = Argmax_{A \in S(\mathcal{C})}(m(A)) \tag{7.33}$$

The assignment of a_i is precise when A^* is a singleton and imprecise when A^* is a set of successive categories. Let us note that this rule can produce multiple maxima, each leading to a particular assignment. In such a situation, all the possible assignments should be proposed to the decision maker.

A comparison of DISSET with an ELECTRE TRI like procedure has been proposed in Boujelben *et al.* (2007). The results have shown an important degree of compatibility between these approaches especially when the size of the LS rises. The robustness of DISSET has also been analyzed in Boujelben *et al.* (2007). It has been proved to be robust to changes in the initial labels. Unfortunately, DISSET has not yet been applied to real world problems. This will certainly constitute a next development.

7.4.4 A choice model inspired by ELECTRE I

In addition to the multicriteria approaches described above, evidence theory has also been used in the modeling of a choice method inspired by ELECTRE I in the multi-experts context (Boujelben *et al.* 2009a). Briefly, ELECTRE I (Roy 1968) is a multicriteria outranking approach that consists of choosing a subset of actions considered as the best according to the criteria set. The underlying idea of this method is first to build a binary relation denoted S between pairs of alternatives and then to exploit this relation to select the subset of 'best' actions.

The model inspired by ELECTRE I supposes that the actions are evaluated on a set of ordinal criteria and that the evaluations given by the experts are imperfect. Moreover, it considers that the information related to the criteria weights is imprecise. For instance, one could accept that the total importance allocated to the coalition of financial criteria, including investment costs and operational costs, is equal to 0.6 while the importance related to each of the single criteria is not determined or is such that their sum is lower than 0.6.

The concept of BBA has been used in this method to model both the imperfect evaluations and the imprecise criteria weights. In what follows, let $m^h_{i|f}$ be the BBA describing the evaluation of action a_i on criterion g_h and given by expert E_f and m_G be the BBA defining the criteria weights. Intuitively, $m_G(T)$ (where $T \subseteq G$) represents the weight committed exactly to the criteria set T (i.e., the weight that the decision maker does not assign to any element of T). $Bel_G(T)$ represents the total weight assigned to T. The concept of BBA allows therefore the definition of weights on single criteria and on coalitions of criteria. Of course, m_G is a weighted sum when it is a Bayesian BBA. Let us note that a similar concept which is the Choquet capacity (Choquet 1953) is also used in MCDA to represent the criteria weights. This approach has been used mainly in MAUT (Angilella *et al*. 2004; Grabisch 1996).

In order to select the subset of the 'best' actions, the model inspired by ELECTRE I proceeds in four steps. At first, each expert has to compare each pair of BBAs characterizing the evaluations on each criterion. This is performed using the concept of first belief dominance described in Section 7.3.1. This approach allows concluding if the evaluation of an action a_i dominates the evaluation of an action $a_{i'}$ on each criterion or not, i.e., if a_i is at least as good as $a_{i'}$ on each criterion or not (while integrating the belief information). Once all the belief dominance relations between the pairs of actions on each criterion are determined, they are exploited in the next step to build the binary outranking relations.

The construction of an outranking relation S between two actions a_i and $a_{i'}$ by each expert E_f requires computing two degrees: the concordance degree and the incomparability degree. The first one measures the degree to which a_i is at least as good as $a_{i'}$, i.e., the importance of criteria subset $D_{i,i'|f}$ defined such as:

$$D_{i,i'|f} = \left\{ g_h \in G;\ m^h_{i|f} \text{ FBD } m^h_{i'|f} \right\} \tag{7.34}$$

The second one quantifies the degree to which a_i and $a_{i'}$ are incomparable (i.e., $m^h_{i|f}$ and $m^h_{i'|f}$ do not dominate each other), i.e., the importance of criteria subset $Q_{i,i'|f}$ defined such as:

$$Q_{i,i'|f} = \left\{ g_h \in G;\ m^h_{i|f}\ \overline{\text{FBD}}\ m^h_{i'|f} \text{ and } m^h_{i'|f}\ \overline{\text{FBD}}\ m^h_{i|f} \right\} \tag{7.35}$$

Since the criteria weights are expressed by the BBA m_G, the concordance and incomparability degrees are therefore the belief degrees $Bel_G(D_{i,i'|f})$ and $Bel_G(Q_{i,i'|f})$. The outranking relation S is built by comparing these degrees with concordance and incomparability thresholds denoted, respectively, c and r and which are fixed by the decision maker. a_i is said to outrank $a_{i'}$ (i.e., $a_i S a_{i'}$) according to expert E_f if and only if

$Bel_G(D_{i,i'|f})$ is greater than or equal to c and $Bel_G(Q_{i,i'|f})$ is lesser than or equal to r. More formally:

$$a_i S a_{i'} \Leftrightarrow \begin{cases} Bel_G\left(D_{i,i'|f}\right) \geq c \\ Bel_G\left(Q_{i,i'|f}\right) \leq r \end{cases} \tag{7.36}$$

Once each expert determines all the outranking relations between the pairs of actions, he establishes his individual outranking graph synthesizing these relations.

In the third step of the model, the individual outranking graphs are aggregated in order to establish a collective one on which the decision is based. This is performed using the algorithm AL3 proposed by Jabeur and Martel (2002). This procedure is based on the notion of distance between preference relations (Jabeur *et al*. 2004) and takes into account the coefficients of the experts' importance within the group. It determines, for each pair of alternatives a_i and $a_{i'}$ the nearest collective preference relation $R^* \in \{P, P^{-1}, I, J\}$ to the individual ones [where P is the preference relation (P^{-1} for the inverse), I the indifference relation and J the incomparability relation]. For this purpose, a divergence degree $\Phi^R(a_i, a_{i'})$ that quantifies the deviation between the collective preference relation $R \in \{P, P^{-1}, I, J\}$ and each individual one $R^f(a_i, a_{i'})$ is calculated as follows:

$$\Phi^R(a_i, a_{i'}) = \sum_{f=1}^{s} w_f . \Delta(R, R^f(a_i, a_{i'})) \tag{7.37}$$

where w_f is the importance of expert E_f within the group and $\Delta(R, R^f(a_i, a_{i'}))$ is the distance measure between relations R and $R^f(a_i, a_{i'})$ (the numerical values of the distance measure are given in Table 7.2). Then, the collective relation $R^*(a_i, a_{i'})$ that minimizes the divergence degrees is identified as follows:

$$R^* = Argmin_{R\in\{P,P^{-1},I,J\}} \Phi^R(a_i, a_{i'}) \tag{7.38}$$

Once the collective preference relations between all pairs of alternatives are determined, we build the collective graph.

In the last step of the model, the collective outranking graph is exploited to select the 'best' alternatives. This is based on the notion of graph kernel (Vincke 1992).

Finally, let us note that the application of the algorithm AL3 may produce several collective graphs and therefore different subsets of 'best' alternatives can be deduced. Moreover, let us stress that the interactive and iterative procedure developed by Jabeur and Martel can be used to help the experts to reach a consensus on the 'best' alternatives

Table 7.2 Numerical values of $\Delta(R, R')$ (Jabeur *et al*. 2004).

	I	P	J	P^{-1}
I	$\Delta(I, I) = 0$	$\Delta(I, P) = 1$	$\Delta(I, J) = 4/3$	$\Delta(I, P^{-1}) = 1$
P	$\Delta(P, I) = 1$	$\Delta(P, P) = 0$	$\Delta(P, J) = 4/3$	$\Delta(P, P^{-1}) = 5/3$
J	$\Delta(J, I) = 4/3$	$\Delta(J, P) = 4/3$	$\Delta(J, J) = 0$	$\Delta(J, P^{-1}) = 4/3$
P^{-1}	$\Delta(P^{-1}, I) = 1$	$\Delta(P^{-1}, P) = 5/3$	$\Delta(P^{-1}, J) = 4/3$	$\Delta(P^{-1}, P^{-1}) = 0$

subset if they do not accept the result given by the algorithm AL3. Further details can be found in Jabeur and Martel (2005).

7.4.5 A ranking model inspired by Xu *et al.*'s method

Recently, evidence theory has been used to model a ranking procedure inspired by Xu *et al.*'s method in the multi-experts context (Boujelben *et al.* 2011). Briefly, Xu *et al.*'s approach (Xu *et al.* 2001) is a ranking model which uses partial or total preorders of the actions on each criterion. These unicriterion preorders reflect situations of preference P_h, (P_h^{-1} for the inverse), indifference I_h and incomparability J_h between the actions on each criterion g_h. The underlying idea of this method is that an action a_i performs better if there are more relations $a_i P_h a_{i'}$ and fewer relations $a_i P_h^{-1} a_{i'}$ for $i' \neq i$. Following this intuition, Xu *et al.* have proposed to use the notion of distance between preference relations mentioned earlier (see Table 7.2) to quantify the performance of each action a_i. More precisely, they have used the distances between P_h and $R_h(a_i, a_{i'})$ denoted $\Delta(P_h, R_h(a_i, a_{i'}))$ and between P_h^{-1} and $R_h(a_i, a_{i'})$ denoted $\Delta(P_h^{-1}, R_h(a_i, a_{i'}))$ (where $R_h(a_i, a_{i'}) \in \{P_h, P_h^{-1}, I_h, J_h\}$). Of course, the less $\Delta(P_h, R_h(a_i, a_{i'}))$ and the more $\Delta(P_h^{-1}, R_h(a_i, a_{i'}))$, the better a_i. These distances allow, respectively, building two total preorders O_1 and O_2 which are then combined to determine the global ranking of the actions denoted O.

As the choice model described above, the ranking model inspired by Xu *et al.*'s method supposes that the actions are evaluated on a set of ordinal criteria and that the evaluations are imperfect and modeled by BBAs. Furthermore, contrary to the evidential reasoning algorithm, the assessment grades set used to evaluate the actions is not the same for all the criteria, i.e., each criterion has its own set of evaluation grades. In what follows, let $m_{i|f}^h$ be the BBA that represents the evaluation of action a_i according to criterion g_h and given by expert E_f, w_f be the importance of expert E_f within the group and $w_{h|f}$ be the weight of criterion g_h given by expert E_f.

In order to rank the alternatives, the model inspired by Xu *et al.*'s method proceeds in three steps. At first, the BBAs given by each expert and characterizing the actions performances on each criterion are ranked using the RBBD concept described in Section 7.3.2. Two unicriterion preorders of the evaluations can be deduced: the RBBD I and II rankings. As mentioned before, the former gives a partial preorder of the evaluations on each criterion according to each expert and allows taking into account situations of incomparability. The latter gives a total preorder of the evaluations on each criterion according to each expert.

In the second step of the model, the unicriterion preorders of the evaluations determined by each expert are aggregated in order to determine the individual global ranking of the actions. The aggregation is performed as in Xu *et al.*'s method. For that purpose, each expert determines at first the two individual total preorders O_1^f and O_2^f. Let $R_h^f(a_i, a_{i'}) \in \{P_h, P_h^{-1}, I_h, J_h\}$ be the preference relation between a_i and $a_{i'}$ observed on the unicriterion preorder of g_h given by expert E_f. The preorder O_1^f is built iteratively using the distance that quantifies the individual dominating character of a_i on all criteria defined as follows:

$$d^{O_1^f}(a_i) = \sum_{h=1}^{q} w_{h|f} . d_f^{P_h}(a_i) \tag{7.39}$$

where $d_f^{P_h}(a_i) = \sum_{i' \neq i} \Delta(P_h, R_h^f(a_i, a_{i'}))$ is a distance that describes the individual dominating character of a_i on criterion g_h. Of course, the less the distance $d^{O_1^f}(a_i)$, the better a_i. Based on the values of $d^{O_1^f}(a_i)$, the best action(s) is (are) determined, i.e., the action(s) that has (have) the minimum of these values. Then, the above distances are recomputed but only for the actions that are not yet ranked. These distances are determined without taking into account the action(s) already ranked. Then, the best alternative(s) is (are) determined as previously. This procedure continues as described above until we obtain the preorder O_1^f. In the same way, the preorder O_2^f is established iteratively on the basis of the distance representing the individual dominated character of a_i on all the criteria and given by the following formula:

$$d^{O_2^f}(a_i) = \sum_{h=1}^{q} w_{h|f}.d_f^{P_h^{-1}}(a_i) \tag{7.40}$$

where $d_f^{P_h^{-1}}(a_i) = \sum_{i' \neq i} \Delta(P_h^{-1}, R_h^f(a_i, a_{i'}))$ is a distance that describes the individual dominated character of a_i on g_h. Then, the two total preorders O_1^f and O_2^f are combined in order to obtain the individual global ranking of the actions O^f. This is performed using the following rules:

- $a_i P a_{i'}$ in O^f if $a_i P a_{i'}$ in both O_1^f and O_2^f, or if $a_i P a_{i'}$ in one preorder and $a_i I a_{i'}$ in the other;
- $a_i I a_{i'}$ in O^f if $a_i I a_{i'}$ in both O_1^f and O_2^f;
- $a_i J a_{i'}$ in O^f if $a_i P a_{i'}$ in one preorder and $a_i P^{-1} a_{i'}$ in the other.

In the last step of the model, the individual global rankings are aggregated in order to obtain the collective global ranking of the actions. This is based on the same procedure used in Xu *et al.*'s method in the aggregation of the unicriterion preorders. We should therefore determine at first the two collective total preorders O_1 and O_2. In what follows, let $R^f(a_i, a_{i'}) \in \{P, P^{-1}, I, J\}$ be the preference relation between a_i and $a_{i'}$ observed on the individual global ranking O^f.

The collective total preorder O_1 is built iteratively using a distance that quantifies the collective dominating character of a_i according to all experts. More formally:

$$d^{O_1}(a_i) = \sum_{f=1}^{s} w_f.d_f^P(a_i) \tag{7.41}$$

where $d_f^P(a_i) = \sum_{i' \neq i} \Delta(P, R^f(a_i, a_{i'}))$ is a distance that quantifies the individual dominating character of a_i according to expert E_f. The distance $d^{O_1}(a_i)$ is computed at each iteration for the actions not yet ranked and without considering the actions that are ranked in the previous steps. Similarly, the collective total preorder O_2 is built iteratively, but instead of considering the collective dominating character of a_i according to all experts,

we consider its collective dominated intensity measured by the following distance:

$$d^{O_2}(a_i) = \sum_{f=1}^{s} w_f \cdot d_f^{P^{-1}}(a_i) \tag{7.42}$$

where $d_f^{P^{-1}}(a_i) = \sum_{i' \neq i} \Delta(P^{-1}, R^f(a_i, a_{i'}))$ is a distance that quantifies the individual dominated character of a_i according to expert E_f. The collective global preorder O is then determined as the intersection of O_1 and O_2 using the same rules described above.

Finally, let us note that at the second step of the model, each expert has the possibility to choose between the two unicriterion preorders of the evaluations (i.e., the RBBD I or II rankings) before applying the aggregation procedure of Xu *et al.*'s method. The choice between them can also be imposed by the decision maker.

7.5 Discussion

The previous section has been devoted to a description of multicriteria methods modeled by evidence theory. In what follows, we will stress a few general observations about the application of this theory in the field of MCDA.

- **New concepts within evidence theory for MCDA**. First of all, evidence theory offers several concepts that can be applied directly to MCDA. However, if we consider order relations which are omnipresent in MCDA, new approaches should be defined in order to take them into account. For this purpose, two recent concepts in evidence theory which are the first belief dominance and the RBBD approach have been developed. Let us recall that the former allows pairwise comparisons between BBAs whereas the latter permits the BBAs to be ranked. These concepts have been used respectively to compare evaluations expressed by BBAs in the choice model inspired by ELECTRE I and to rank evaluations represented by BBAs in the ranking model inspired by Xu *et al.*'s method. We are convinced that other new concepts will be developed in the near future to facilitate the application of evidence theory to MCDA.
- **Modeling the imperfection in data**. Evidence theory offers convenient tools to tackle imperfection in data in MCDA. In particular, the concept of BBA allows experts to express freely imperfect evaluations of the actions on ordinal criteria. These evaluations are determined on the basis of a set of assessment grades that can be defined for all the criteria as in the evidential reasoning approach or for each criterion as in the models inspired by ELECTRE I and Xu *et al.*'s methods. The set of assessment grades constitutes in evidence theory the frame of discernment. However, the distinctive feature of this frame is related to the fact that ordinal information is available about its elements. At this point, let us note that a direct consequence of considering a frame consisting of ordered elements is that the focal elements should be either singletons or disjunctions of successive elements.

  The concept of BBA has also been applied in DISSET to model comparisons between the action to be assigned and the alternatives from the LS and in DS/AHP

to express the priority values of sets of actions on each criterion. Moreover, it has been used in the choice model inspired by ELECTRE I to represent the criteria weights when the information about them is imprecise. Indeed, the notion of BBA allows defining weights on single criteria and even on coalitions of criteria. As it has been mentioned before, this function permits the decision maker to assign exactly weights to single criteria or sets of criteria and its related belief function allows total weights of coalitions of criteria to be represented.

- **Combination.** In addition to its ability to tackle imperfect data, the combination rules offered by evidence theory can be solutions to aggregate information in multicriteria decision problems. For instance, Dempster's rule has been used in the evidential reasoning algorithm to combine the BBAs characterizing the evaluations of a given action in order to obtain a global evaluation. It has also been applied in DS/AHP to aggregate the BBAs representing the priority values of sets of actions on each criterion in order to obtain global priority values. The use of this rule in DS/AHP has also been extended in the group decision making context in order to determine collective global priority values. Furthermore, Dempster's rule has been used in DISSET to aggregate the BBAs deduced from the comparison of the action to be assigned to the alternatives. The resulting BBA is then exploited to assign this action to its appropriate category or categories. Let us note that in all these cases, only Dempster's rule has been applied since the BBAs to combine are assumed to be independent.

  However, the use of combination rules offered by evidence theory can be limited in some multicriteria decision problems. Indeed, as one can suggest in the choice model inspired by ELECTRE I, another manner for the aggregation of the different experts' opinions is the use of Dempster's rule (the cautious rule, respectively) to combine the BBAs given by independent (dependent, respectively) experts and that represent the evaluations of action a_i on criterion g_h. Let m_i^h be the BBA resulting from the combination of the BBAs given by the experts (i.e., $m_{i|1}^h$, $m_{i|2}^h$, ... and $m_{i|s}^h$). This BBA represents of course the collective evaluation of action a_i according to criterion g_h. Once all collective BBAs are determined, the first belief dominance approach is used to compare these BBAs and to build the collective outranking graph of which the best alternatives subset is identified. However, Dempster's or the cautious rules do not respect in some situations the unanimity property which means that:

  — If $m_{i|f}^h$ FBD $m_{i'|f}^h$ for all $f \in \{1, \ldots, s\}$, then m_i^h FBD $m_{i'}^h$;

  — If $m_{i|f}^h$ $\overline{FBD}$ $m_{i'|f}^h$ for all $f \in \{1, \ldots, s\}$, then m_i^h $\overline{\text{FBD}}$ $m_{i'}^h$.

  That is why the algorithm AL3 has been used in the aggregation step. Similarly, in the ranking model inspired by Xu *et al.*'s method, it is possible to use Dempster's rule or the cautious rule in order to determine for each action a_i the BBA m_i^h representing its collective evaluation according to the criterion g_h. Once all the collective BBAs are determined, the RBBD concept is used to rank them. Two collective unicriterion preorders of the evaluations can be deduced: the RBBD I and II rankings. These preorders (RBBD I or II) are then aggregated as in Xu *et al.*'s method in order to obtain the collective global ranking of the actions.

However, Dempster's and the cautious rules do not respect also in some situations the unanimity property. This means that:

— If $m_{i|f}^{h} P_h m_{i'|f}^{h}$ for all $f \in \{1, \ldots, s\}$, then $m_i^h P_h m_{i'}^h$;

— If $m_{i|f}^{h} P_h^{-1} m_{i'|f}^{h}$ for all $f \in \{1, \ldots, s\}$, then $m_i^h P_h^{-1} m_{i'}^h$;

— If $m_{i|f}^{h} I_h m_{i'|f}^{h}$ for all $f \in \{1, \ldots, s\}$, then $m_i^h I_h m_{i'}^h$;

— If $m_{i|f}^{h} J_h m_{i'|f}^{h}$ for all $f \in \{1, \ldots, s\}$, then $m_i^h J_h m_{i'}^h$.

That is why the procedure used in Xu *et al*.'s method has been applied in the aggregation step. Of course, the development of combination rules that respect the unanimity principle will be a crucial research question for the successful application of evidence theory to MCDA.

7.6 Conclusion

In this chapter, we have been interested in applications of evidence theory in the field of MCDA. Our objective has been to highlight interactions between these two disciplines (that have been developed separately). In this spirit, we have reviewed five multicriteria methods modeled by evidence theory and we have illustrated how evidence theory has been applied in the main multicriteria problematics. As mentioned earlier, this theory offers appropriate tools to represent imperfect data and to combine information in the context of multicriteria analysis. The evidential reasoning approach, DS/AHP and DISSET can be viewed as direct applications of this theory since they are modeled using its traditional notions (BBA, Dempster's rule, etc.). In addition, two other recent concepts which are the first belief dominance approach and the RBBD concept have been proposed for MCDA which allow dealing, respectively, with the problem of comparing and ranking BBAs. These two concepts have been used for modeling a choice method inspired by ELECTRE I and a ranking procedure inspired by Xu *et al*.'s approach.

At this point, we have illustrated the benefits of using evidence theory in the field of MCDA. Of course, there are still many directions for future research. Among others, we can mention the development of combination rules that respect the unanimity property according to the first belief dominance and the RBBD approach. Moreover, as stressed earlier, the concept of BBA has been used to represent imperfect evaluations according to ordinal criteria. An interesting line of research at this level is to develop procedure(s) allowing the elicitation of BBAs reflecting well the judgments of the experts. The elicitation problem is central in MCDA. Using tools developed in this context to assess belief distributions could be an interesting application of MCDA to evidence theory.

References

Angilella S, Greco S, Lamantia F and Matarazzo B (2004) Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research* **158**, 734–744.

Appriou A (1991) Probabilités et incertitude en fusion de données multisenseurs. *Revue Scientifique et Technique de la Défense* **11**, 27–40.

Augusto M, Lisboa JV, Yasin MM and Figueira J (2008) Benchmarking in a multiple criteria performance context: an application and a conceptual framework. *European Journal of Operational Research* **184**, 244–254.

Belacel N (2000) Multicriteria assignment method PROAFTN: Methodology and medical applications. *European Journal of Operational Research* **125**, 175–183.

Beynon M (2002) DS/AHP method: A mathematical analysis including an understanding of uncertainty. *European Journal of Operational Research* **140**, 148–164.

Beynon M (2005a) A method of aggregation in DS/AHP for group decision-making with the non-equivalent importance of individuals in the group. *Computers & Operations Research* **32**, 1881–1896.

Beynon M (2005b) Understanding local ignorance and non-specificity within the DS/AHP method of multi-criteria decision making. *European Journal of Operational Research* **163**, 403–417.

Beynon M, Cosker D and Marshall D (2001) An expert system for multi-criteria decision making using Dempster–Shafer theory. *Expert Systems with Applications* **20**, 357–367.

Beynon M, Curry B and Morgan P (2000) The Dempster–Shafer theory of evidence: an alternative approach to multicriteria decision modelling. *Omega* **28**, 37–50.

Biswas G, Oliff M and Sen A (1988) An expert decision support system for production control. *Decision Support Systems* **4**, 235–248.

Boujelben MA (2011) *On the use of evidence theory in multi-criteria decision aid*. PhD thesis, Faculté des Sciences Economiques et de Gestion de Sfax.

Boujelben MA, De Smet Y, Frikha A and Chabchoub H (2007) DISSET: a disjunctive sorting method based on evidence theory. *Foundations of Computing and Decision Sciences* **32**, 253–274.

Boujelben MA, De Smet Y, Frikha A and Chabchoub H (2009a) Building a binary outranking relation in uncertain, imprecise and multi-experts contexts: the application of evidence theory. *International Journal of Approximate Reasoning* **50**, 1259–1278.

Boujelben MA, De Smet Y, Frikha A and Chabchoub H (2009b) The first belief dominance: a new approach in evidence theory for comparing basic belief assignments. In *ADT'09: Proceedings of the 1st International Conference on Algorithmic Decision Theory*. Springer-Verlag, Berlin, pp. 272–283.

Boujelben MA, De Smet Y, Frikha A and Chabchoub H (2011) Ranking model in uncertain, imprecise and multi-experts contexts: the application of evidence theory. *International Journal of Approximate Reasoning* **8**, 1171–1194.

Brans JP and Mareschal B (2005) PROMETHEE methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueria J, Greco S and Ehrgott M). Springer-Verlag, Boston, pp. 163–189.

Brans JP and Vincke P (1985) A preference ranking organisation method: the PROMETHEE method for MCDM. *Management Science* **31**, 647–656.

Choquet G (1953) Theory of capacities. *Annales de l'Institut Fourier* **5**, 131–296.

Dempster A (1967) Upper and lower probabilities induced by a multi-valued mapping. *Annual Mathematics and Statistics* **38**, 325–339.

Deng H (1999) Multicriteria analysis with fuzzy pairwise comparison. *International Journal of Approximate Reasoning* **21**, 215–231.

Denoeux T (2008) Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence* **172**, 234–264.

Denoeux T (2009) Extending stochastic ordering to belief functions on the real line. *Information Sciences* **179**, 1362–1376.

Denoeux T and Masson MH (2004) EVCLUS: Evidential clustering of proximity data. *IEEE Transactions on Systems, Man and Cybernetics B* **34**, 95–109.

Denoeux T and Smets P (2006) Classification using belief functions: the relationship between the case-based and model-based approaches. *IEEE Transactions on Systems, Man and Cybernetics B* **36**, 1395–1406.

Doumpos M and Zopounidis C (2002) Multicriteria classification methods in financial and banking decisions. *International Transactions in Operational Research* **9**, 567–581.

Dubois D and Prade H (1980) *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York.

Dubois D and Prade H (1988a) *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York.

Dubois D and Prade H (1988b) Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence* **4**, 244–264.

Figueira J, Greco S and Ehrgott M (eds) (2005a) *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer-Verlag, Boston.

Figueira J, Mousseau V and Roy B (2005b) ELECTRE methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueria J, Greco S and Ehrgott M). Springer-Verlag, Boston, pp. 133–162.

Fishburn PC (1970) *Utility Theory for Decision-making*. John Wiley & Sons, Ltd, New York.

Grabisch M (1996) The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research* **89**, 445–456.

Guo M, Yang JB, Chin KS and Wang H (2007) Evidential reasoning based preference programming for multiple attribute decision analysis under uncertainty. *European Journal of Operational Research* **182**, 1294–1312.

Hadar J and Russell WR (1969) Rules for ordering uncertain prospects. *The American Economic Review* **59**, 25–34.

Ha-Duong M (2008) Hierarchical fusion of expert opinions in the Transferable Belief Model: application to climate sensitivity. *International Journal of Approximate Reasoning* **49**, 555–574.

Hwang C and Yoon K (1981) *Multiple Attribute Decision Making: Methods and Applications*. New York, Springer.

Hyde KM, Maier HR and Colby C (2003) Incorporating uncertainty in the PROMETHEE MCDA method. *Journal of Multi-Criteria Decision Analysis* **12**, 245–259.

Jabeur K and Martel JM (2002) Détermination déun (ou plusieurs) systèmes(s) relationnel(s) de préférence (s.r.p) collectif(s) à partir des s.r.p individuels. Document de travail no. 011-2002, Faculté des Sciences de lÁdministration (FSA), Université Laval.

Jabeur K and Martel JM (2005) A collective choice method based on individual preferences relational systems (p.r.s.). *European Journal of Operational Research* **177**, 469–485.

Jabeur K, Martel JM and Ben Khélifa S (2004) A distance-based collective pre-order integrating the relative importance of the group's members. *Group Decision and Negotiation* **13**, 327–349.

Janez F (1996) *Fusion de sources d'information définies sur des référentiels non exhaustifs différents: solutions proposées sous le formalisme de la théorie de lévidence*, PhD thesis, Université d'Angers.

Keeney R and Raiffa H (1976) *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley & Sons, Ltd, New York.

Masson MH and Denoeux T (2008) ECM: an evidential version of the fuzzy c-means algorithm. *Pattern Recognition* **41**, 1384–1397.

Murphy C (2000) Combining belief functions when evidence conflicts. *Decision Support Systems* **29**, 1–9.

Roy B (1968) Classement et choix en présence de points de vue multiples: La méthode ELECTRE. *Revue Franąise d'Informatique et de Recherche Opérationnelle* **8**, 57–75.

Roy B (1989) Main sources of inaccurate determination, uncertainty and imprecision. *Mathematical and Computer Modelling* **12**, 1245–1254.

Roy B and Bouyssou D (1993) *Aide multicritère à la décision: méthodes et cas*. Economica, Paris.

Saaty T (1980) *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill Inc., New York.

Shafer G (1976) *A Mathematical Theory of Evidence*. Princeton University Press, Princeton.

Smets P (1988) Belief functions. In *Non Standard Logics for Automated Reasoning* (ed. Smets P), Academic Press, London, pp. 253–286.

Smets P (1990) *Constructing the pignistic probability function in a context of uncertainty*. In *Uncertainty in Artificial Intelligence* (eds Henrion M, Lemmer JF, Kanal LN and Shachter RD). Elsevier Science, Amsterdam, pp. 29–40.

Smets P (2002) Decision making in a context where uncertainty is represented by belief functions. In *Belief Functions in Business Decisions* (eds Srivastava RP and Mock TJ). Physica-Verlag, Heidelberg, pp. 17–61.

Smets P (2005) Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning* **38**, 133–147.

Smets P (2007) Analyzing the combination of conflicting belief functions. *Information Fusion* **8**, 387–412.

Smets P and Kennes R (1994) The transferable belief model. *Artificial Intelligence* **66**, 191–234.

Sönmez M, Yang JB and Hol GD (2001) Addressing the contractor selection problem using an evidential reasoning approach. *Engineering Construction and Architectural Management* **8**, 198–210.

Vincke P (1992) *Multicriteria Decision-Aid*. John Wiley & Sons, Ltd, New York.

Wang J and Yang JB (2001) A subjective safety based decision making approach for evaluation of safety requirements specifications in software development. *International Journal of Reliability Quality and Safety Engineering* **8**, 35–57.

Wang YM, Yang JB and Xu DL (2006) The evidential reasoning approach for multiple attribute decision analysis using interval belief degrees. *European Journal of Operational Research* **175**, 35–66.

Williams D (1991) *Probability with Martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge.

Xu L, Krzyzak A and Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics* **22**, 418–435.

Xu X, Martel JM and Lamond BF (2001) A multiple criteria ranking procedure based on distance between partial preorders. *European Journal of Operational Research* **133**, 69–80.

Yager RR (1987) On the Dempster–Shafer framework and new combination rules. *Information Sciences* **41**, 93–137.

Yager RR and Liu L (2008) *Classic Works of the Dempster–Shafer Theory of Belief Functions*. Springer-Verlag, Berlin.

Yamada K (2008) A new combination of evidence based on compromise. *Fuzzy Sets and Systems* **159**, 1689–1708.

Yang JB (2001) Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties. *European Journal of Operational Research* **131**, 31–61.

Yang JB and Sen P (1997) Multiple attribute design evaluation of large engineering products using the evidential reasoning approach. *Journal of Engineering Design* **8**, 211–230.

Yang JB and Singh MG (1994) An evidential reasoning approach for multiattribute decision making with uncertainty. *IEEE Transactions on Systems, Man and Cybernetics* **24**, 1–18.

Yang JB and Xu DL (2002) On the evidential reasoning algorithm for multiattribute decision analysis under uncertainty. *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans* **32**, 289–304.

Yang JB, Wang YM, Xu DL and Chin KS (2006) The evidential reasoning approach for MCDA under both probabilistic and fuzzy uncertainties. *European Journal of Operational Research* **171**, 309–343.

Yu W (1992) ELECTRE TRI: Aspects méthodologiques et manuel d'utilisation. *Document du LAMSADE no. 74*, Université Paris-Dauphine, Paris, France.

Zadeh L (1965) Fuzzy sets. *Information and Control* **8**, 338–353.

Zadeh L (1978) Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems* **1**, 3–28.

Zavadskas EK, Turskis Z, Tamosaitiene J and Marina V (2008) Multicriteria selection of project managers by applying grey criteria. *Technological and Economic Development of Economy, Baltic Journal on Sustainability* **14**, 462–477.

Zopounidis C and Doumpos M (2002) Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research* **138**, 229–246.

Part IV

MULTIOBJECTIVE OPTIMIZATION

8

Interactive approaches applied to multiobjective evolutionary algorithms

Antonio López Jaimes, and Carlos A. Coello Coello
CINVESTAV-IPN, Departamento de Computación, Evolutionary Computation Group (EVOCINV), Mexico

8.1 Introduction

Multiobjective evolutionary algorithms (MOEAs) rely on preference relations to steer the search towards high-potential regions of the search space in order to approximate the optimal solution set. In particular, a preference relation is a means to decide if a solution is preferable over another solution in the search space.

In single-objective optimization, the determination of the optimum among a set of given solutions is clear. However, in the absence of preference information, in multiobjective optimization, there does not exist a unique preference relation to determine if a solution is better than another one. The most common preference relation adopted is known as the *Pareto dominance relation* (Pareto 1896), which leads to the best possible trade-offs among the objectives. Thus, by using this relation, it is normally not possible to obtain a single optimal solution (except when there is no conflict among the objectives), but instead, a set of good solutions (representing the best possible trade-offs among the objectives) can be produced. This set is called the *Pareto optimal set* and its image in objective space is known as the *Pareto optimal front*.

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

Multiobjective optimization involves three stages: model building, search, and decision making (preference articulation). Having a good approximation of the Pareto optimal set does not completely solve a multiobjective optimization problem. The decision maker (DM) still has the task of choosing the most preferred solution out of the approximation set. This task requires preference information from the DM. Following this need, there are several methodologies available for defining how and when to incorporate preferences from the DM into the search process. These methodologies can be classified in the following categories (Coello Coello *et al*. 2007; Miettinen 1998):

(1) Prior to the search (a priori approaches).

(2) During the search (interactive approaches).

(3) After the search (a posteriori approaches).

Although interactive approaches for incorporating preferences have been widely used for a long time in Operations Research (see e.g., Chankong and Haimes 1983; Miettinen 1998), it was only until very recently that the inclusion of preference information into MOEAs started to attract a considerable amount of interest among researchers (see e.g., Branke 2008; Coello Coello *et al*. 2007). Regardless of the stage at which preferences are incorporated into a MOEA, the aim is to focus on a certain portion of the Pareto front by favoring certain objectives (or trade-offs) over others.

As noted by several researchers (Hughes 2005; Khare *et al*. 2003; Knowles and Corne 2007; Praditwong and Yao 2007; Purshouse and Fleming 2007; Teytaud 2007; Wagner *et al*. 2007), the Pareto dominance relation has an important drawback when it is applied to multiobjective optimization problems with a high number of objectives (these are the so-called *many-objective problems*, e.g., Kukkonen and Lampinen 2007). The drawback is the deterioration of its ability to discern between good and bad solutions as the number of solutions increases. A widely accepted explanation for this problem is that the proportion of nondominated solutions (i.e., incomparable solutions according to the Pareto dominance relation) in a population increases rapidly with the number of objectives (see, e.g., Bentley *et al*. 1978; Farina and Amato 2002). Since incorporating preferences induces a finer order on vectors of the objective space than that achieved by the Pareto dominance relation, we believe that the use of the new preference relation is a promising approach to deal with many-objective problems. Additionally, by using an interactive optimization technique we can avoid the generation of millions or even billions of nondominated points in many-objective problems.

This chapter presents a review of recent MOEAs designed to work as interactive optimization methods. For earlier methods, the reader is referred to other specialized reviews such as those presented by Branke and Deb (2005), Coello Coello (2000), Cvetković and Parmee (2002), and Rachmawati and Srinivasan (2006). The contents of this chapter aims to complement these previous reviews of the field instead of aiming at being comprehensive.

8.1.1 Methods analyzed in this chapter

By analyzing the approaches found in other reviews and the ones covered in this chapter, we can realize that most of the approaches to incorporate preferences into MOEAs are based on methods introduced in the field of multicriteria decision making. For instance, we

can find many approaches based on reference point methods. There are a few methods that originate from the evolutionary multiobjective optimization (EMO) field or come from other areas. We can mention, for example, a method based on the hypervolume indicator. Thus, the methods analyzed in this chapter are classified into the following categories:

- reference point methods;
- utility function methods;
- miscellaneous methods.

The survey is mainly focused on interactive MOEAs. However, some a priori techniques can be easily set for selecting the location and size of the region of interest (ROI). This way, they can serve as a basis for interactive approaches. Therefore, we also include some of these interesting a priori techniques for incorporating preferences.

8.2 Basic concepts and notation

In this section, we will introduce the concepts and notation that will be used throughout the rest of the chapter. Furthermore, as many interactive MOEAs are based on classical interactive methods proposed by the Operations Research community, some of these methods are first described.

8.2.1 Multiobjective optimization problems

Definition 8.2.1 (Multiobjective optimization problem) *A multiobjective optimization problem (MOP) is defined as:*

$$\begin{aligned} &\textit{Minimize} \quad \mathrm{f}(\mathrm{x}) = [f_1(\mathrm{x}), f_2(\mathrm{x}), \ldots, f_k(\mathrm{x})]^T, \\ &\textit{subject to} \quad \mathrm{x} \in \mathcal{X}. \end{aligned} \tag{8.1}$$

The vector $\mathrm{x} \in \mathbb{R}^n$ is formed by n *decision variables* representing the quantities for which values are to be chosen in the optimization problem. The *feasible set* $\mathcal{X} \subseteq \mathbb{R}^n$ is implicitly determined by a set of equality and inequality constraints. The vector function $\mathrm{f} : \mathcal{X} \to \mathbb{R}^k$ is composed of $k \geq 2$ scalar *objective functions* $f_i : \mathcal{X} \to \mathbb{R}$ $(i = 1, \ldots, k)$. In multiobjective optimization, the sets $\mathbb{R}^n$ and $\mathbb{R}^k$ are known as *decision variable space* and *objective function space*, respectively. The image of $\mathcal{X}$ under the function f is a subset of the objective function space denoted by $\mathcal{Z} = \mathrm{f}(\mathcal{X})$ and referred to as the *feasible set in the objective function space*.

In order to define precisely the multiobjective optimization problem stated in Definition 8.2.1 we have to establish the meaning of minimization in $\mathbb{R}^k$. That is to say, we need to define how vectors $\mathrm{z} = \mathrm{f}(\mathrm{x}) \in \mathbb{R}^k$ have to be compared for different solutions $\mathrm{x} \in \mathbb{R}^n$. In single-objective optimization the relation 'less than or equal' ($\leq$) is used to compare the scalar objective values. By using this relation there may be many different optimal solutions $\mathrm{x} \in \mathcal{X}$, but only one optimal value $f^{\min} = \min\{f(x) \mid x \in \mathcal{X}\}$ since the relation $\leq$ induces a total order in $\mathbb{R}$ (i.e., every pair of solutions is comparable, and thus, we can sort solutions from the best to the worst one). In contrast, in MOPs,

there is no canonical order on $\mathbb{R}^k$, and thus, we need weaker definitions of order to compare vectors in $\mathbb{R}^k$.

In multiobjective optimization, the *Pareto dominance relation* is usually adopted. This relation was originally proposed by Francis Ysidro Edgeworth (1881), and was generalized by the French-Italian economist Vilfredo Pareto (1896).

Definition 8.2.2 (Pareto dominance relation) *We say that a vector* z^1 *dominates vector* z^2*, denoted by* $\mathrm{z}^1 \prec \mathrm{z}^2$*, if and only if:*

$$\forall i \in \{1, \ldots, k\} : z_i^1 \leq z_i^2 \tag{8.2}$$

and

$$\exists i \in \{1, \ldots, k\} : z_i^1 < z_i^2. \tag{8.3}$$

If $\mathrm{z}^1 = \mathrm{z}^2$ or $z_i^1 > z_i^2$ for some i, then we say that z^1 does not dominate z^2 (denoted by $\mathrm{z}^1 \nprec_{\text{pareto}} \mathrm{z}^2$). Thus, to solve a MOP we have to find those solutions $\mathrm{x} \in \mathcal{X}$ whose images, $\mathrm{z} = \mathrm{f}(\mathrm{x})$, are not dominated by any other vector in the feasible space. It is said that two vectors, z^1 and z^2, are *mutually nondominated vectors* if $\mathrm{z}^1 \nprec_{\text{pareto}} \mathrm{z}^2$ and $\mathrm{z}^2 \nprec_{\text{pareto}} \mathrm{z}^1$.

Definition 8.2.3 (Pareto optimality) *A solution* $\mathrm{x}^* \in \mathcal{X}$ *is Pareto optimal if there does not exist another solution* $\mathrm{x} \in \mathcal{X}$ *such that* $\mathrm{f}(\mathrm{x}) \prec \mathrm{f}(\mathrm{x}^*)$.

Definition 8.2.4 (ρ-properly Pareto optimality) *A solution* $\mathrm{x}^* \in \mathcal{X}$ *and its corresponding vector* $\mathrm{z}^* \in \mathcal{Z}$ *are ρ-properly Pareto optimal (in the sense of Wierzbicki (1980b)) if*

$$(\mathrm{z}^* - \mathbb{R}^k_\rho \backslash \{0\}) \cap \mathcal{Z} = \emptyset,$$

where $\mathbb{R}^k_\rho = \{\mathrm{z} \in \mathbb{R}^k | \max_{i=1,\ldots,k} z_i + \rho \sum_{i=1}^k z_i \geq 0\}$, *and ρ is some scalar. The trade-offs among the objectives are bounded by ρ and $1/\rho$.*

Definition 8.2.5 (Pareto optimal set) *The Pareto optimal set,* P_{opt}, *is defined as:*

$$P_{\text{opt}} = \{\mathrm{x} \in \mathcal{X} \mid \nexists\, \mathrm{y} \in \mathcal{X} \;:\; \mathrm{f}(\mathrm{y}) \prec \mathrm{f}(\mathrm{x})\}. \tag{8.4}$$

Definition 8.2.6 (Pareto front) *For a Pareto optimal set,* P_{opt}, *the Pareto front,* PF_{opt}, *is defined as:*

$$PF_{\text{opt}} = \{\mathrm{z} = (f_1(\mathrm{x}), \ldots, f_k(\mathrm{x})) \mid \mathrm{x} \in P_{\text{opt}}\}. \tag{8.5}$$

In decision variable space, these vectors are referred to as decision vectors of the Pareto optimal set, while in objective space, they are called objective vectors of the Pareto optimal set. In practice, the goal of a posteriori approaches is finding the 'best' approximation set of the Pareto optimal front. An approximation set is a finite subset of $\mathcal{Z}$ composed of mutually nondominated vectors and is denoted by PF_{approx}. Currently, it is well accepted that the best approximation set is determined by the closeness to the

Pareto optimal front, and the spread over the entire Pareto optimal front (Coello Coello *et al*. 2007; Deb *et al*. 2002b; Zitzler *et al*. 2003).

In interactive optimization methods it is useful to know the lower and upper bounds of the Pareto front. The *ideal point*, $\mathrm{z}^\star$, represents the lower bounds and is defined by $z_i^\star = \min_{\mathrm{z}\in\mathcal{Z}}\{z_i\}\ \forall i = 1,\ldots,k$. In turn, the upper bounds are defined by the *nadir point*, z^{nad}, which is given by $z_i^{\mathrm{nad}} = \max_{\mathrm{z}\in P} F_{\mathrm{opt}}\{z_i\}\ \forall i = 1,\ldots,k$. In order to avoid some problems when the ideal and nadir points are equal or very close, a point strictly better than the ideal point is usually defined. This point is called the *utopian point*, $\mathrm{z}^{\star\star}$, and is defined by $z_i^{\star\star} = z_i^\star - \varepsilon$, $\forall i = 1, \ldots, k$, where $\varepsilon > 0$ is a small scalar.

8.2.2 Classical interactive methods

8.2.2.1 Reference point methods

These kinds of methods are based on the achievement scalarizing function approach proposed by Wierzbicki (1980a,b). An achievement scalarizing function uses a reference point to capture the desired values of the objective functions.

Definition 8.2.7 (Achievement scalarizing function) *An achievement scalarizing function (or achievement function for short) is a parameterized function $s_{\mathrm{z}^{\mathrm{ref}}}(\mathrm{z}) : \mathbb{R}^k \to \mathbb{R}$, where $\mathrm{z}^{\mathrm{ref}} \in \mathbb{R}^k$ is a reference point representing the decision maker's aspiration levels. Thus, the multiobjective problem is transformed into the following scalar problem:*

$$\begin{aligned} &\textit{Minimize} && s_{\mathrm{z}^{\mathrm{ref}}}(\mathrm{z}), \\ &\textit{subject to} && \mathrm{z} \in \mathcal{Z}. \end{aligned} \tag{8.6}$$

A common achievement function is based on the Chebyshev distance (L_∞ metric) (see e.g., Ehrgott 2005; Miettinen 1998).

Definition 8.2.8 (Chebyshev distance) *For two vectors $\mathrm{z}^1, \mathrm{z}^2 \in \mathbb{R}^k$ the Chebyshev distance is defined by*

$$d_\infty(\mathrm{z}^1, \mathrm{z}^2) = \|\mathrm{z}^1 - \mathrm{z}^2\|_\infty = \max_{i=1,\ldots,k} |z_i^1 - z_i^2|. \tag{8.7}$$

Definition 8.2.9 (Weighted achievement function) *The weighted achievement function (or achievement function for short) is defined by*

$$s_\infty(\mathrm{z}, \mathrm{z}^{\mathrm{ref}}) = \max_{i=1,\ldots,k} \{\lambda_i(z_i - z_i^{\mathrm{ref}})\} + \rho \sum_{i=1}^{k} \lambda_i(z_i - z_i^{\mathrm{ref}}), \tag{8.8}$$

where $\mathrm{z}^{\mathrm{ref}}$ is a reference point, $\lambda = [\lambda_1, \ldots, \lambda_k]$ is a vector of weights such that $\forall i\ \lambda_i \geq 0$ and, for at least one i, $\lambda_i > 0$, and $\rho > 0$ is a sufficiently small augmentation coefficient (usually $\rho = 10^{-6}$). The main role of ρ is to avoid the generation of weakly Pareto optimal solutions.

We should note that, unlike the Chebyshev distance, the achievement function does not use the absolute value in the first term. This small difference allows the achievement function to correctly assess solutions that improve the reference point.

The achievement function has some convenient properties over other scalarizing functions. As proved, for instance by Steuer (1986), Miettinen (1998) and Ehrgott (2005), the minimum of Equation (8.8) is a Pareto optimal solution and we can find any ρ-properly Pareto optimal solution (see Definition 8.2.4).

8.2.2.2 Light beam search method

The Light Beam Search (LBS) method proposed by Jaszkiewicz and Slowinski (1999), is an iterative method which combines the reference point idea and tools of multi-attribute decision analysis (MADA). At each iteration, a finite sample of nondominated points is generated. The sample is composed of a current point called the *middle point* (which is obtained at a previous iteration), and J nondominated points from its neighborhood. A local preference model in the form of an *outranking relation* S is used to define the neighborhood of the middle point. It is said that a vector z^1 outranks vector z^2 ($\mathrm{z}^1 S \mathrm{z}^2$) if z^1 is considered to be at least as good as z^2. The outranking relations are defined by the DM, which specify three preference thresholds for each objective, namely: *indifference threshold, preference threshold* and *veto threshold*. The DM has the possibility to scan the inner area of the neighborhood along the objective function trajectories between any two characteristic neighbors or between a characteristic neighbor and the middle point. In Algorithm 8.1 the general scheme of the LBS procedure is shown.

8.3 MOEAs based on reference point methods

8.3.1 A weighted distance metric

Deb and Sundar (2006) incorporated a reference point approach into the Nondominated Sorting Genetic Algorithm-II (NSGA-II) (Deb *et al*. 2002a). They introduced a modification in the crowding distance operator in order to select from the last nondominated front the solutions that would take part of the new population. They used the following achievement function based on the Euclidean distance

$$d(\mathrm{z}, \mathrm{z}^{\mathrm{ref}}) = \sqrt{\sum_{i=1}^{k} w_i \left(\frac{z_i - z_i^{\mathrm{ref}}}{z_i^{\max} - z_i^{\min}} \right)^2}, \tag{8.9}$$

where $\mathrm{z} \in \mathbb{R}^k$ is a solution, $\mathrm{z}^{\mathrm{ref}}$ is a reference point, $z_i^{\max} = \max_{\mathrm{z}\in P}\{z_i\}$ and $z_i^{\min} = \min_{\mathrm{z}\in P}\{z_i\}$ $\forall i = 1, \ldots, k$, calculated with respect to the current population, P, and the weight vector should satisfy $w_i \in [0, 1]$ and $\sum_{i=1}^{M} w_i = 1$.

The value of this function was used to sort and rank the population accordingly (the solution closest to the reference point receives the best rank). In order to control the spread of the ROI, solutions whose achievement value differs by an amount of ε or less receive the same rank. This way, a set of solutions clustered around the best ranked solution forms the ROI. This method was designed to take into account a set of reference points, i.e., several independent ROIs can be generated. A drawback of this scheme is that it might generate some non-Pareto optimal solutions, particularly in MOPs with disconnected Pareto fronts. Furthermore, we want to point out that the location of the reference point and shape of the Pareto front also determine the size of the ROI and, in

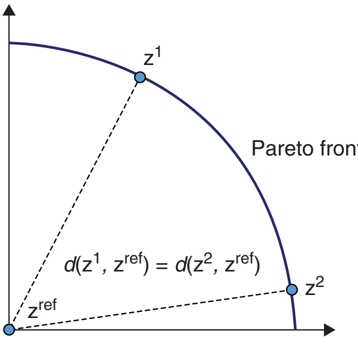

Figure 8.1 The location of the reference point and the shape of the Pareto front might avoid reducing the region of interest.

general, the order in which solutions are ranked. Let us take for example the Pareto front of the 2-objective DTLZ2 problem (Figure 8.1). If we choose the ideal point as reference point, then for any value of ε, all the solutions in the Pareto front will be equally ranked since they are equidistant to the origin (e.g., solutions $\mathbf{z}^1$ and $\mathbf{z}^2$ in Figure 8.1).

Algorithm 8.1 (General scheme of the Light Beam Search procedure)

Step 1: Ask the (DM) to specify the starting aspiration and reservation points.
Step 2: Compute the starting middle point.
Step 3: Ask the DM to specify a local preferential information used to build an outranking relation.
Step 4: Present the middle point to the DM.
Step 5: Calculate the characteristic neighbors of the middle point and present them to the DM.
Step 6: **If** DM is satisfied **then**
  Stop.
else

6.1: Ask DM to choose one of the neighboring points to be the new middle point, or

6.2: Update the preferential information, or

6.3: Define new aspiration point and/or reservation point.

6.4: Go to **Step 3**.

An interesting case is observed when the reference point is farther away from the origin. In that case, solutions in the extreme of the Pareto front are closer to the origin than the solution in the center of the Pareto front. This means that the DM should have some knowledge about the shape and lower bounds of the Pareto front in order to avoid these situations.

8.3.2 Light beam search combined with NSGA-II

A similar approach was also proposed by Deb and Kumar (2007), in which the LBS procedure (Jaszkiewicz and Slowinski 1999) was incorporated into NSGA-II. Similar to the previous approach, they modified the crowding operator to incorporate the DM's preferences. They used a weighted achievement function to assign a crowding distance to each solution in each front. Thus, the solution with the least distance will have the best crowding rank. As in the previous approach, this algorithm finds a subset of solutions around the optimum of the achievement function adopting the outranking relation proposed by Jaszkiewicz and Slowinski (1999). In Jaszkiewicz and Slowinski (1999) three kinds of thresholds are defined to determine if one solution outranks another one. However, in Deb and Kumar (2007) the veto threshold is the only one used. This relation depends on the crowding comparison operator.

Although these two previous techniques were presented and implemented as an a priori technique, it is possible to formulate an interactive method with the same basic idea.

8.3.3 Controlling the accuracy of the Pareto front approximation

Kaliszewski *et al*. (2012) presented an interesting interactive approach coupled with a MOEA. The main contribution of this work is a mechanism to control the accuracy of the subset of nondominated solutions obtained. The authors adopted an achievement function to generate a set of weakly Pareto optimal solutions. In order to control the accuracy of the Pareto front approximation, two sets are generated and updated during the search: a set of feasible nondominated solutions, S_L, and a set of infeasible nondominated solutions, A_U, such that $S_L \not\prec A_U$ (Figure 8.2). The basic idea is to enclose the Pareto front with two approximation sets, one approaching from below and another one from above. The accuracy of the Pareto front approximation is determined by a distance measure between S_L and A_U.

The DM's preferences are expressed by means of the so-called *vector of concessions* which represents proportions in which the DM agrees to sacrifice unattainable values of the objectives represented by the ideal point in the hope of getting Pareto optimal solutions.

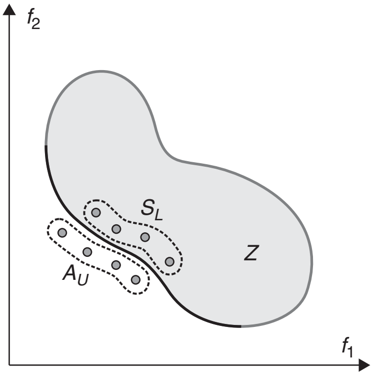

Figure 8.2 Controlling accuracy of the Pareto front approximation.

8.3.4 Light beam search combined with PSO

Wickramasinghe and Li (2009) proposed the combination of the LBS method (Jaszkiewicz and Slowinski 1999) and a particle swarm optimization (PSO) technique to guide the swarm towards a ROI according to the aspiration and reservation points provided by the DM. The main idea is ranking the population using the outranking relation instead of the usual Pareto dominance. By changing the threshold parameters of the outranking relations (indifference threshold, preference threshold, veto threshold) the size of the ROI near the point that minimizes Equation (8.8) can be regulated. The authors used the Multi-objective Differential Evolution and Particle Swarm Optimization (MDEPSO) algorithm as a framework to incorporate LBS. The key points in which LBS is inserted are the following:

- Each time a new particle is created its achievement function value is calculated.
- The leaders to guide the population are sorted according to its achievement function value using the outranking relation. This is the step in which the spread of the ROI is controlled.
- The personal best of each particle will take the value of the updated position only if the achievement function value is improved.
- In order to obtain the new population, the original and updated particles are mixed and sorted according to the outranking relation. Finally, the best half of the mixed population will form the new population.

8.3.5 A preference relation based on a weighted distance metric

Said *et al*. (2010) use the achievement function given by Equation (8.9) to create a new preference relation called r-dominance. This relation combines the usual Pareto dominance and the achievement function in the following way.

Definition 8.3.1 *Given a set of solutions* P *and a reference point* $\mathrm{z}^{\mathrm{ref}}$*, a solution* z^1 *is said to* r*-dominate a solution* z^2 *if:*

(1) $\mathrm{z}^1 \prec \mathrm{z}^2$*, or*

(2) z^1 *and* z^2 *are mutually non dominated solutions, and* $D(\mathrm{z}^1, \mathrm{z}^2, \mathrm{z}^{\mathrm{ref}}) < -\delta$*, where* $\delta \in [0, 1]$ *and*

$$D(\mathrm{z}^1, \mathrm{z}^2, \mathrm{z}^{\mathrm{ref}}) = \frac{d(\mathrm{z}^1, \mathrm{z}^{\mathrm{ref}}) - d(\mathrm{z}^2, \mathrm{z}^{\mathrm{ref}})}{Dist_{\max} - Dist_{min}},$$

$$Dist_{\max} = \max_{\mathrm{z}\in P} d(\mathrm{z}, \mathrm{z}^{\mathrm{ref}}),$$

$$Dist_{\max} = \min_{\mathrm{z}\in P} d(\mathrm{z}, \mathrm{z}^{\mathrm{ref}}),$$

The authors prove that the r-dominance relation, in the same way as the Pareto dominance, defines a strict partial order on a set of solutions since the relation is irreflexive,

asymmetric and transitive. They also prove that r-dominance is complete with the Pareto dominance and compatible with the non-Pareto dominance, i.e., if z^1 r-dominates z^2, then $z^1 \prec z^2$, and if z^1 r-dominates z^2, then $z^2 \nprec_{par} z^1$. In spite of these desirable properties, the r-dominance relation has some drawbacks due to the Euclidean distance function adopted. The size of the ROI is mainly determined by the threshold δ. If $\delta = 1$, r-dominance is equivalent to the Pareto dominance relation. In turn, if $\delta = 0$, in most cases, the relation becomes more stringent since only those solutions minimizing Equation (8.8) will be the most preferred by the relation.

The r-dominance relation was inserted in the interactive method presented in Algorithm 8.2 using a variant of NSGA-II as the search engine.

Algorithm 8.2 (Interactive optimization using the r-dominance relation)

Step 1: Ask the DM for the following parameter values: population size, number of generations, reference solution, weight vector and threshold δ.
Step 2: Apply r-NSGA-II the number of generations required.
Step 3: Present to the DM the set of preferred solutions.
Step 4: **If** the DM is satisfied with the provided set of solutions, **then**
  Stop the process.
 Otherwise
  Ask the DM for new values for the parameters and return to **Step 2**

8.3.6 The Chebyshev preference relation

This preference relation, proposed by López-Jaimes *et al*. (2011), is based on the Chebyshev achievement function [see Equation (8.8)], and provides a simple way to integrate preferences into different types of MOEAs. The basic idea of the Chebyshev preference relation is to combine the Pareto dominance relation and an achievement function to compare solutions in objective function space.

First, the achievement function value, $s_\infty(z, z^{ref})$, is computed for each solution z. Then, the objective space is divided into two regions. One region defines the ROI and contains those solutions with an achievement value less or equal to $s^{min} + \delta$, where $s^{min} = \min_{z \in \mathcal{Z}} s_\infty(z, z^{ref})$, and δ is a threshold that determines the size of the ROI. Figure 8.3 shows the ROI defined by means of the achievement function. Solutions in this region are compared using the usual Pareto dominance relation, while solutions outside of the ROI are compared using their achievement function value.

Formally, the Chebyshev preference relation is defined in the following way.

Definition 8.3.2 *A solution* z^1 *is preferred to solution* z^2 *with respect to the Chebyshev relation* ($z^1 \prec_{cheby} z^2$)*, if and only if:*

(1) $s_\infty(z^1, z^{ref}) < s_\infty(z^2, z^{ref}) \wedge \{z^1 \notin R(z^{ref}, \delta) \vee z^2 \notin R(z^{ref}, \delta)\}$*, or,*

(2) $z^1 \prec z^2 \wedge \{z^1, z^2 \in R(z^{ref}, \delta)\}$,

where $R(z^{ref}, \delta) = \{z \mid s_\infty(z, z^{ref}) \leq s^{min} + \delta\}$ *is the region of interest with respect to the vector of aspiration levels* z^{ref}.

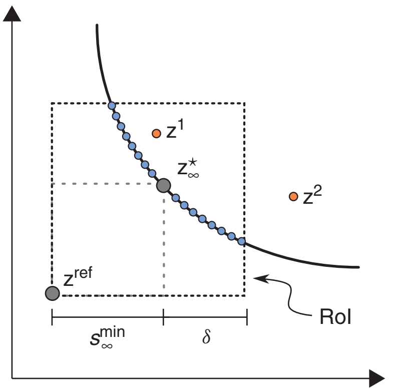

Figure 8.3 Nondominated solutions with respect to the Chebyshev relation.

The threshold δ is set in terms of the user parameter $\tau \in [0, 1]$ according to $\delta = \tau \cdot (s^{\max} - s^{\min})$, where $s^{\max} = \max_{z \in P} s_\infty(\mathrm{z}, \mathrm{z}^{\mathrm{ref}})$ and $s^{\min} = \min_{z \in P} s_\infty(\mathrm{z}, \mathrm{z}^{\mathrm{ref}})$. In this way, if $\tau = 1$, all the solutions in the population P are compared adopting the usual Pareto dominance relation. On the other hand, if $\tau = 0$, then all the solutions are compared using the achievement function value.

Unlike some distance metrics, the achievement function [Equation (8.8)] allows a MOEA to find points in problems with nonconvex Pareto fronts.

López-Jaimes *et al*. (2011) also proposed a variant of the Chebyshev relation that uses an approximation of the ideal point as reference point in Definition 8.3.2. This variant is called the *central-guided Chebyshev relation* since it focuses the search towards the ideal point.

Algorithm 8.3 shows the interactive process using the Chebyshev relation.

Algorithm 8.3 (Interactive technique using the Chebyshev preference relation)

Step 1: Ask the DM to specify the threshold τ.
If the DM has some knowledge about the problem, he/she can provide a reference point. Otherwise, the central-guided preference relation can be used to converge towards the ideal point.

Step 2: **If** a reference point was provided, **then**
  Execute the {MOEA} using the Chebyshev relation with the reference point provided by the DM.
else
  Execute the MOEA using the central-guided Chebyshev relation.

Step 3: Ask the DM to define how many solutions of the current approximation should be shown.
Additionally, from the use of the central-guided relation the DM can be informed of the current ideal point in order to decide new aspiration levels.

Step 4: **If** the DM is satisfied with some solution of the current set, **then**
  STOP.
else
  Go to **Step 1**

8.4 MOEAs based on value function methods

8.4.1 Progressive approximation of a value function

Deb *et al*. (2010) proposed a method in which the DM's value function is progressively approximated through pairwise comparisons of a small set of solutions.

After applying a MOEA, a selection of well-distributed solutions of the achieved Pareto front approximation is presented to the DM (in the experiments reported by the authors, five solutions are shown). Then, for every pair of solutions the DM should establish which one is preferred over the other, or if they are incomparable. Based on this preference information a polynomial value function, $V(\mathrm{z})$, is built. For 2-objective problems, this function is the product of two linear functions whose parameters must be determined using the preference information as constraints of an optimization problem. The created value function is employed to define a preference relation in order to select the parents and, later, to decide which individuals will survive to the next generation. This preference relation uses the value function value (denoted by V_2) of the second-best solution found in the population.

Definition 8.4.1 *A solution* z^1 *is preferred to solution* z^2 *with respect to a value function value* V_2 *if and only if:*

(1) $\mathrm{z}^1 \prec \mathrm{z}^2$*, or*

(2) $V(\mathrm{z}^1) > V_2$ *and* $V(\mathrm{z}^2) < V_2$.

The value function is also used to formulate a termination criterion of the interactive process. The idea is to perform a linear search along the gradient of the value function taking the best solution in the population as initial point. If this solution is improved by a given threshold value, the interactive process continues and the MOEA is applied again. Otherwise, the process stops and the solution found in the linear search is considered the most preferred solution.

8.4.2 Value function by ordinal regression

For a given class of value function it is possible that many of its instances generated by changing the parameters can be compatible with the provided preference information. In many approaches, like the one presented in Section 8.4.1, only one specific instance is used to evaluate the set of solutions. However, as Branke *et al*. (2010) pointed out, since this selection is rather arbitrary, a more robust approach should take into consideration all the set of value functions compatible with the preference information (Figure 8.4). This motivation was the origin of the robust ordinal regression (ROR) proposed by Branke *et al*. (2010). When all the compatible value function are considered, two different preference relations can be defined:

(1) Necessary preference relation: a solution z^1 is ranked at least as good as z^2 if z^1 is preferred over z^2 in all compatible instances of the value function.

(2) Possible preference relation: a solution z^1 is ranked at least as good as z^2 if z^1 is preferred over z^2 in at least one compatible instance of the value function.

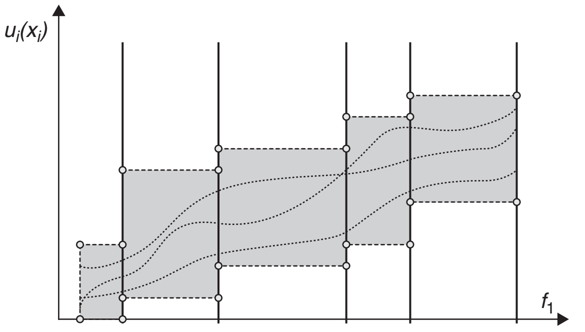

Figure 8.4 Range of compatible value functions for objective f_i.

The necessary preference relation is robust in the sense that any pair of solutions is compared the same whatever the compatible instance of the value function.

In order to define a necessary preference ranking, the preference information is obtained by asking the DM to make a pairwise comparison of a small set of alternative solutions. Then, the necessary preference relation is computed by solving a linear programming problem. In some decision-making situations it is useful to know the most representative value function among all the compatible ones. The authors considered the most representative value function as the one which maximizes the difference of scores between alternatives related by preference in the necessary ranking.

The concept of ROR was integrated into NSGA-II making two important changes:

(1) The Pareto dominance is replaced by the necessary preference relation in such a way that the selection for reproduction and the selection for survival are carried out according to the necessary preference rank.

(2) The crowding distance is computed taking into account the multidimensional scaling given by the most representative value function.

During the main loop of NSGA-II, after k generations, the DM is asked for new preference information.

8.5 Miscellaneous methods

8.5.1 Desirability functions

Wagner and Trautmann (2010) proposed transforming each objective of the problem by a desirability function (DF) that maps the original objective to a domain [0, 1]. This new function has a bias towards the preferred solutions according to the desired values provided by the DM.

Let us consider that the image of objective f_i is $\mathcal{Z}_i \subseteq \mathbb{R}$, then a DF is defined as any function $d_i : \mathcal{Z}_i \rightarrow [0, 1]$ that specifies the desirability of different regions of the domain $\mathcal{Z}_i$ for objective f_i. The authors adopted two types of DF which were introduced by Harrington (1965): one designed for maximization or minimization of the objectives (one-sided function), and another one for target value problems (two-sided function).

Thus, the original MOP is transformed into

$$\begin{aligned} \text{Minimize} & -d(\mathrm{z}) = -d[\mathrm{f}(\mathrm{x})] = -(d_1[f_1(\mathrm{x})], \ldots, d_k[f_k(\mathrm{x})])^T \\ \text{where} \quad & \mathrm{x} \in \mathcal{X}, \\ & \mathrm{z} \in \mathcal{Z}, \\ & d(\mathrm{z}) : \mathcal{Z} \to [0, 1], \; i = 1, \ldots, k. \end{aligned} \tag{8.10}$$

Since the transformed problem is also a MOP, the multi-objective optimization algorithm adopted to solve the original problem can be used without modification to solve the new problem. However, the Pareto optimal solutions of the modified problem will present a biased distribution towards the ROI of the original MOP. For the one-sided DF, the DM states his/her preferences by setting two points of each objective's DF: the first point, $(z_i^{(1)}, d_i^{(1)})$ represents the most desirable value, $d_i^{(1)} = 1$, in the domain $\mathcal{Z}_i$, while the second point, $(z_i^{(2)}, d_i^{(2)})$, denotes the least desirable value, $d_i^{(2)} = 0$, in the domain. The values for $z_i^{(1)}$ and $z_i^{(2)}$ are taken from the range $[z_i^\star, z_i^{\text{nad}}]$, i.e., the range of the Pareto optimal front. For example, in order to focus the search on the center of a Pareto front, $z_i^{(1)} = z_i^\star$ and $z_i^{(2)} = z_i^{\text{nad}}/2$ should be used.

8.6 Conclusions and future work

This chapter has presented a short review of recent efforts to design interactive MOEAs. It is clear that most of the proposals are based on classical well-known techniques that originated in the OR field. From the proposed interactive methods, the most popular approach is the reference point method. This opens important paths for future research. There are many interactive techniques proposed by the OR community and, therefore, one of the obvious future research paths is the development of new interactive MOEAs using techniques based on a classification of the objectives, trade-off methods, or marginal rates of substitution. Another interesting possibility would be the development of interactive MOEAs using concepts from other sources. For example, the incorporation of preferences through the bias of the hypervolume [or other indicators used for assessing performance of MOEAs (Zitzler *et al*. 2003)].

The use of interactive MOEAs to deal with problems with a high number of objectives has become a popular research trend within evolutionary multi-objective optimization. We believe that there are two main reasons for this. On the one hand, the incorporation of preferences avoids the problem of visualizing a huge number of solutions in high dimensionality. On the other hand, emphasizing a ROI introduces a stringent criterion that allows the comparison of nondominated solutions.

Summarizing, we believe that the incorporation of preferences into MOEAs is a very important research topic, not only because this is a fundamental part of the decision-making process involved in the solution of a MOP, but also because it can help to deal with problems having a large number of objectives. This topic, however, is still scarcely researched in the current literature, mainly because of its strong links with OR, which makes it necessary to have a good background in such a discipline as well as in EMO. However, as more effort is being made to bring together these two communities (OR and EMO) (Branke *et al*. 2008) we expect to see much more research in this area in the next few years.

Acknowledgment

The authors acknowledge support from CONACyT project no. 103570.

References

Bentley JL, Kung HT, Schkolnick M and Thompson CD (1978) On the average number of maxima in a set of vectors and applications. *Journal of the ACM* **25**(4), 536–543.

Branke J (2008) Consideration of partial user preferences in evolutionary multiobjective optimization. In *Multiobjective Optimization. Interactive and Evolutionary Approaches* (eds Branke J, Deb K, Miettinen K and Slowinski R). Lecture Notes in Computer Science, vol 5252. Springer, Berlin, pp. 157–178.

Branke J and Deb K (2005) Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge Incorporation in Evolutionary Computation* (ed. Jin Y). Springer, Berlin, pp. 461–477.

Branke J, Greco S, Slowinski R and Zielniewicz P (2010) Interactive evolutionary multiobjective optimization driven by robust ordinal regression. *Bulletin of the Polish Academy of Sciences-Technical Sciences* **58**(3), 347–358.

Chankong V and Haimes YY (1983) *Multiobjective decision making: theory and methodology*. North-Holland, New York, NY.

Coello Coello CA (2000) Handling preferences in evolutionary multiobjective optimization: A survey. *2000 Congress on Evolutionary Computation*, vol. 1. IEEE Service Center, Piscataway, NJ, pp. 30–37.

Coello Coello CA, Lamont GB and Van Veldhuizen DA (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd. Springer, New York.

Cvetković D and Parmee IC (2002) Preferences and their application in evolutionary multiobjective optimisation. *IEEE Transactions on Evolutionary Computation* **6**(1), 42–57.

Deb K and Kumar A (2007) Light beam search based multi-objective optimization using evolutionary algorithms *IEEE Congress on Evolutionary Computation*, pp. 2125–2132.

Deb K and Sundar J (2006) Reference point based multi-objective optimization using evolutionary algorithms. In *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, pp. 635–642.

Deb K, Pratap A, Agarwal S and Meyarivan T (2002a) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Deb K, Sinha A, Korhonen PJ and Wallenius J (2010) An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation* **14**(5), 723–739.

Deb K, Thiele L, Laumanns M and Zitzler E (2002b) Scalable multi-objective optimization test problems. *Congress on Evolutionary Computation (CEC'-2002)*, vol. 1. IEEE Service Center, Piscataway, NJ, pp. 825–830.

Branke J, Deb K, Miettinen K and Slowinski R (eds) (2008) *Multiobjective Optimization. Interactive and Evolutionary Approaches*. Lecture Notes in Computer Science, vol 5252. Springer, Berlin.

Edgeworth FY (1881) *Mathematical Physics*. P. Keagan, London.

Ehrgott M (2005) *Multicriteria Optimization*, 2nd edn. Springer, Berlin.

Farina M and Amato P (2002) On the optimal solution definition for many-criteria optimization problems. In *Proceedings of the NAFIPS-FLINT International Conference' 2002*. IEEE Service Center, Piscataway, NJ, pp. 233–238.

Harrington EC (1965) The desirability function. *Industrial Quality Control* **21**(10), 494–498.

Hughes EJ (2005) Evolutionary many-objective optimisation: Many once or one many? *2005 IEEE Congress on Evolutionary Computation (CEC'-2005)*, vol. 1. IEEE Service Center, Edinburgh, pp. 222–227.

Jaszkiewicz A and Slowinski R (1999) The Light Beam Search approach – an overview of methodology and applications. *European Journal of Operational Research* **113**(2), 300–314.

Kaliszewski I, Miroforidis J and Podkopaev D (2012) Interactive Multiple Criteria Decision Making based on preference driven Evolutionary Multiobjective Optimization with controllable accuracy. *European Journal of Operational Research* **216**(1), 188–199.

Khare V, Yao X and Deb K (2003) Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (eds Fonseca CM, Fleming PJ, Zitzler E, Deb K and Thiele L). Lecture Notes in Computer Science, vol. 2632. Springer, Berlin, pp. 376–390.

Knowles J and Corne D (2007) Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In *volutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (eds Obayashi S, Deb K, Poloni C, Hiroyasu T and Murata T). Lecture Notes in Computer Science, vol. 4403. Springer, Berlin, pp. 757–771.

Kukkonen S and Lampinen J (2007) Ranking-dominance and many-objective optimization. *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*. IEEE Press, Singapore, pp. 3983–3990.

López-Jaimes A, Arias-Montaño A and Coello Coello CA (2011) Preference incorporation to solve many-objective airfoil design problems. *2011 IEEE Congress on Evolutionary Computation (CEC, 2011)*. IEEE Press, New York, NY, pp. 1605–1612.

Miettinen KM (1998) *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, MA.

Pareto V (1896) *Cours D'Economie Politique*. F. Rouge, Lausanne.

Praditwong K and Yao X (2007) How well do multi-objective evolutionary algorithms scale to large problems. *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*. IEEE Press, Singapore, pp. 3959–3966.

Purshouse RC and Fleming PJ (2007) On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Algorithms* **11**(6), 770–784.

Rachmawati L and Srinivasan D (2006) Preference incorporation in multi-objective evolutionary algorithms: A survey. *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. IEEE, Vancouver, BC, pp. 3385–3391.

Said LB, Bechikh S and Ghédira K (2010) The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation* **14**, 801–818.

Steuer RE (1986) *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, Ltd, New York.

Teytaud O (2007) On the hardness of offline multi-objective optimization. *Evolutionary Computation* **15**(4), 475–491.

Wagner T and Trautmann H (2010) Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation* **14**, 688–701.

Wagner T, Beume N and Naujoks B (2007) Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007* (eds Obayashi S, Deb K, Poloni C, Hiroyasu T and Murata T). Lecture Notes in Computer Science, vol. 4403, Springer, Berlin, pp. 742–756.

Wickramasinghe UK and Li X (2009) Using a distance metric to guide PSO algorithms for many-objective optimization. In GECCO '09: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. ACM, New York, NY, pp. 667–674.

Wierzbicki A (1980a) A methodological guide to multiobjective optimization In *Optimization Techniques, Part 1* (eds Iracki K, Malanowski K and Walukiewicz S), vol. 22 of *Lecture Notes in Control and Information Sciences*, pp. 99–123. Springer, Berlin.

Wierzbicki A (1980b) The use of reference objectives in multiobjective optimisation In *Multiple Criteria Decision Making Theory and Application* (ed. G. F and T. G), pp. 468–486 number 177 in *Lecture notes in economics and mathematical systems*. Springer Verlag, Heilderberg, Berlin.

Zitzler E, Thiele L, Laumanns M, Fonseca CM and da Fonseca VG (2003) Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation* **7**(2), 117–132.

9

Generalized data envelopment analysis and computational intelligence in multiple criteria decision making

Yeboon Yun[1] and Hirotaka Nakayama[2]

[1]*Faculty of Environmental and Urban Engineering, Kansai University, Japan*
[2]*Department of Intelligence and Informatics, Konan University, Japan*

9.1 Introduction

Many kinds of practical decision making problems such as engineering design have multiple objectives conflicting with each other. These problems can be formulated as multi-objective optimization problems (MOPs) as follows:

$$\begin{array}{lll} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top & \text{(MOP)} \\ \text{subject to} & \mathbf{x} \in X = \left\{ \mathbf{x} \in \mathbb{R}^n \mid g_j(\mathbf{x}) \leqq 0, \quad j = 1, \ldots, l \right\}, & \end{array}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^\top$ is a vector of design variables, and X is the set of all feasible solutions (or alternatives).

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

For convenience, the following notation for two vectors $\mathbf{y} = (y_1, \ldots, y_m)^\top$ and $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_m)^\top$ in $\mathbb{R}^m$ will be used:

$$\mathbf{y} > \hat{\mathbf{y}} \quad \Leftrightarrow y_i > \hat{y}_i, \ i = 1, \ldots, m$$
$$\mathbf{y} \geqq \hat{\mathbf{y}} \quad \Leftrightarrow y_i \geqq \hat{y}_i, \ i = 1, \ldots, m$$
$$\mathbf{y} \geq \hat{\mathbf{y}} \quad \Leftrightarrow y_i \geqq \hat{y}_i, \ i = 1, \ldots, m \ \text{ but } \ \mathbf{y} \neq \hat{\mathbf{y}}$$

Generally, there does not exist a unique solution which minimizes all objective functions simultaneously in MOPs. Hence, the concept of an optimal solution based on the relation of Pareto dominance was introduced (Pareto 1906):

Definition 9.1.1 (Pareto optimal solution) *In MOP, a point $\hat{\mathbf{x}} \in X$ is said to be a Pareto optimal solution if there exists no $\mathbf{x} \in X$ such that $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\hat{\mathbf{x}})$. The set of Pareto optimal solutions in the objective function space is called a Pareto frontier. Figure 9.1 shows the Pareto frontier in the case of a problem with two objective functions.*

Therefore, a decision making problem with multiple objectives is to determine the best solution in the set of Pareto optimal solutions. To do this, decision makers should consider the criteria trade-offs, that is some amount should be given up in other objectives if the improvement of some objectives is achieved. To this end, the specification of these trade-offs requires the decision maker to define value judgments, and one of the main tasks of multi-objective optimization is to incorporate the decision makers' value judgments into a decision support process. Multi-attribute utility (value) analysis provides some mathematical form for these value judgments, and interactive multi-objective programming techniques enable the search of solution through the elicitation of partial information on these value judgments.

On the other hand, data envelopment analysis (DEA) uses value judgments to evaluate the efficiency of decision making units (DMUs), which correspond to the alternatives in multiple criteria decision making (MCDM) problems. Thus, seeing efficiencies (or inefficiencies) of alternatives by DEA is helpful in making a final decision in MCDM problems. Various DEA models have been proposed to handle the value judgments of

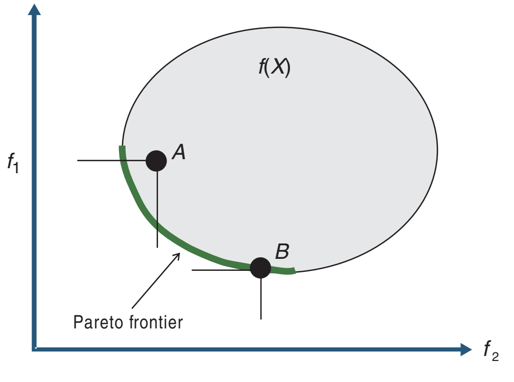

Figure 9.1 Pareto frontier in the objective function space (there is no feasible solution which dominates B, while A is dominated by some feasible solutions).

decision makers, for example the CCR model with ratio (weighted sum) value judgments (Charnes *et al*. 1978; 1979), the BCC model with additive value judgments (Banker *et al*. 1984) and the free disposable hull (FDH)[1] model without introducing any value judgments (Tulkens 1993).

However, in applying DEA to a wide range of practical problems, there are some cases in which those specified value judgments are not adequate. The generalized data envelopment analysis (GDEA) model, which can embed these value judgments in a unified model, was suggested by Yun *et al*. (2004b). The key idea of GDEA is to introduce a domination structure with one parameter varying from the value free structure to a ratio value structure.

In this chapter, we introduce GDEA for practical use in MCDM problems. First, we discuss the relationships between GDEA and existing DEA models, and show that GDEA incorporates various preference structures of decision makers in evaluating DMUs. Utilizing the properties of GDEA, we present several methods to combine GDEA and computational intelligence techniques for generating approximate Pareto optimal solutions which are considered as the candidates of a final solution for MCDM problems. Next, we show that combining GDEA and aspiration level method can make it possible to show the decision makers' most interesting part of Pareto optimal solutions. Particularly, we give an account of the GDEA method in deciding adaptively parameters in multi-objective particle swarm optimization (PSO), and finally, in applying expected improvement (EI) for making good offspring in real-coded genetic algorithms (GAs).

9.2 Generalized data envelopment analysis

DEA was contrived by Charnes *et al*. (1978, 1979) as a method for measuring relative efficiencies of DMUs performing similar tasks in a production system that consumes multiple inputs to produce multiple outputs. This CCR model is the first model in DEA, and the main characteristics of DEA are that:

- it can be applied to analyze the efficiency for multiple outputs and multiple inputs without preassigned weights;
- it can be used for measuring a relative efficiency based on the observed data without knowing information on the production function;
- decision makers' preferences can be incorporated in a DEA model.

Later, Banker *et al*. (1984) suggested a model for distinguishing between technical efficiency and scale inefficiency in DEA. The BCC model relaxes the constant returns to scale assumption of the CCR model, thus enabling the investigation of whether the performance of each DMU is conducted in regions of increasing, constant or decreasing returns to scale in multiple outputs and multiple inputs situations. In addition, Tulkens (1993) introduced an efficiency on the non convex FDH of the observed data. Agrell and Tind (1998) presented the relationships among CCR, BCC, FDH and an MCDA model according to the property of a partial Lagrangean relaxation. Yun *et al*. (2000, 2004b)

[1] The FDH by Deprins *et al.* (1984) is a non convex hull consisting of any points that produce less output with the same amount of input as the observed data, and/or those that require more input with the same amount of output.

suggested a concept of value free efficiency and proposed the GDEA model which can treat basic DEA models (CCR, BCC, FDH) in a unified way. The GDEA model makes it possible to evaluate several efficiencies of DMUs incorporating various preference structures of decision makers.

To begin with, we give an overview of DEA models, and the following notations are used commonly in this section:

- q, m, p: the number of DMUs, inputs and outputs, respectively
- o: an index of DMU to be evaluated
- x_{ik}: ith input of $\text{DMU}_k, k = 1, \ldots, q;\ i = 1, \ldots, m$
- y_{jk}: jth output of $\text{DMU}_k, k = 1, \ldots, q;\ i = 1, \ldots, p$
- ε: sufficiently small positive number (e.g., 10^{-7})
- $\mathbf{1}= (1, \ldots, 1)^\top$.

We assume that $x_{ik} > 0$ for each $i = 1, \ldots, m$ and $y_{jk} > 0$ for each $j = 1, \ldots, p$, and there are no duplicated units in the observed data. The $m \times q$ input matrix for q DMUs is denoted by $\mathbf{X}$, whereas $\mathbf{Y}$ is a $p \times q$ matrix for the outputs of the DMUs. The inputs and outputs for DMU_o are denoted by $\mathbf{x}_o := (x_{1o}, \ldots, x_{mo})^\top$ and $\mathbf{y}_o := (y_{1o}, \ldots, y_{po})^\top$.

9.2.1 Basic DEA models: CCR, BCC and FDH models

For better understanding, we explain DEA models using a simple example of one input and one output as shown in Table 9.1.

The CCR model takes the viewpoint that a scale efficiency is constant, that is, constant returns to scale, and thus the production possibility set becomes a convex cone (or conical hull) generated by the given observed data. On the other hand, assuming varying returns to scale in the BCC model yields that the production possibility set is a convex hull. The FDH model in which the production possibility set is the FDH has no assumption on returns to scale.

Figure 9.2 shows the production possibility set of each DEA model for the data of Table 9.1. The Pareto frontier on the production possibility set is called a DEA-efficient frontier, and DEA measures a relative distance to the DEA-efficient frontier. For example, the efficient value θ_D of DMU_D is given by the ratio of OD' to OD. As seen from Figure 9.2, clearly a DMU on a DEA-efficient frontier has its efficient value $\theta = 1$, and if a DMU is far from the DEA-efficient frontier, the DEA-efficient value is close to 0. Table 9.2 shows DEA-efficient values for the data for Table 9.1 for each DEA model.

Table 9.1 Case of a single input and a single output.

DMU	A	B	C	D	E	F
x (input)	2	4	9	5	7	8
y (output)	2	5	7	3	5.5	4

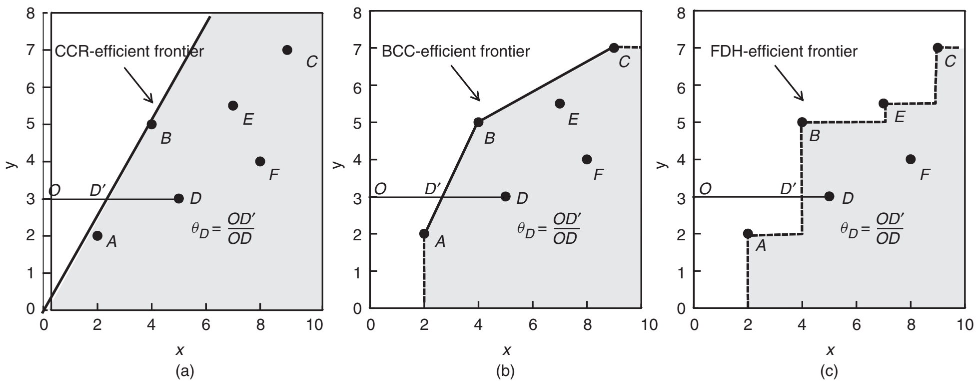

Figure 9.2 DEA-efficient frontiers (the solid line is the efficient frontier, the dotted line is the weak-efficient frontier and the shadowed area is the production possibility set): (a) CCR model; (b) BCC model; (c) FDH model.

Table 9.2 DEA efficiencies for the data in Table 9.1.

DMU	*A*	*B*	*C*	*D*	*E*	*F*
CCR	0.8	1.0	0.62	0.48	0.63	0.40
BCC	1.0	1.0	1.0	0.53	0.75	0.42
FDH	1.0	1.0	1.0	0.80	1.0	0.50

Extending to the case of multiple inputs and multiple outputs, an efficiency can be evaluated by maximizing the following ratio for non-negative input weights $\nu_i, i = 1, \dots, m$ and output weights $\mu_j, j = 1, \dots, p$:

$$\frac{\sum_{j=1}^{p} \mu_j y_{jo}}{\sum_{i=1}^{m} \nu_i x_{io}}. \tag{9.1}$$

Imposing the normality condition that the ratio (9.1) is not larger than one for all DMUs, the CCR model determines optimal weights $\mu = (\mu_i, \dots, \mu_p)^\top, \nu = (\nu_i, \dots, \nu_m)^\top$ and an efficient value θ by solving the following linear programming problem (CCR):

$$\begin{aligned} \underset{\mu,\ \nu}{\text{maximize}} \quad & \theta := \sum_{j=1}^{p} \mu_j y_{jo} && \text{(CCR)} \\ \text{subject to} \quad & \sum_{i=1}^{m} \nu_i x_{io} = 1, \\ & \frac{\sum_{j=1}^{p} \mu_j y_{jk}}{\sum_{i=1}^{m} \nu_i x_{ik}} \leqq 1, k = 1, \dots, q, \end{aligned}$$

$$\mu_j \geqq \varepsilon,\ j = 1, \ldots, p,$$
$$\nu_i \geqq \varepsilon,\ i = 1, \ldots, m.$$

The dual problem (CCR_D) to CCR can be obtained by:

$$\begin{aligned}
&\underset{\theta,\,\boldsymbol{\lambda},\,\mathbf{s_x},\,\mathbf{s_y}}{\text{minimize}} && \theta - \varepsilon\left(\mathbf{1}^\top \mathbf{s_x} + \mathbf{1}^\top \mathbf{s_y}\right) && (\mathrm{CCR}_D)\\
&\text{subject to} && \mathbf{X}\boldsymbol{\lambda} - \theta \mathbf{x}_o + \mathbf{s_x} = 0,\\
& && \mathbf{Y}\boldsymbol{\lambda} - \mathbf{y}_o - \mathbf{s_y} = 0,\\
& && \boldsymbol{\lambda} \geqq \mathbf{0},\ \mathbf{s_x} \geqq 0,\ \mathbf{s_y} \geqq 0,\\
& && \theta \in \mathbb{R},\ \boldsymbol{\lambda} \in \mathbb{R}^q,\ \mathbf{s_x} \in \mathbb{R}^m,\ \mathbf{s_y} \in \mathbb{R}^p.
\end{aligned}$$

Similarly, the BCC model is obtained by adding the constraint $\mathbf{1}^\top\boldsymbol{\lambda} = 1$ to the dual problem (CCR_D), and adding the constraints $\lambda_k \in \{0, 1\}$, $k = 1, \ldots, q$ to the BCC model yields the FDH model. But, the FDH model is a mixed integer programming problem, and hence an efficient value θ is usually obtained by means of a simple vector comparison procedure:

$$\theta := \min_{k \in D(o)} \max_{i=1,\ldots m} \left\{\frac{x_{ik}}{x_{io}}\right\},\ D(o) = \left\{\, j \,|\, \mathbf{x}_k \leq \mathbf{x}_o \ \text{ and } \ \mathbf{y}_k \geqq \mathbf{y}_o, k = 1, \ldots, q \,\right\}.$$

Normally, we give the definition of efficiency by the CCR, BCC, and FDH models as follows:

Definition 9.2.1 (DEA efficiency) *For the optimal solution θ^*, $\mathbf{s}^*_\mathbf{x}$, $\mathbf{s}^*_\mathbf{y}$, DMU is said to be* DEA-efficient *if and only if $\theta^* = 1$, $\mathbf{s}^*_\mathbf{x} = 0$ and $\mathbf{s}^*_\mathbf{y} = 0$.*

The stated models are representative of DEA, and elementary and detailed descriptions of DEA can be found in the book by Cooper *et al*. (2007). In the next subsection, we introduce the GDEA model which can evaluate DEA efficiency in several basic models as special cases, and investigate the relationships between the GDEA model and the DEA models described earlier.

9.2.2 GDEA model

Employing the augmented Tchebyshev scalarizing function, the GDEA model can be formulated as follows:

$$\begin{aligned}
&\underset{\Delta,\,\mu,\,\nu}{\text{maximize}} && \Delta && (\mathrm{GDEA})\\
&\text{subject to} && \Delta \leqq \tilde{d}_k + \alpha\left(\sum_{j=1}^{p} \mu_j (y_{jo} - y_{jk}) + \sum_{i=1}^{m} \nu_i(-x_{io} + x_{ik})\right),\ k = 1, \ldots, q,\\
& && \sum_{j=1}^{p} \mu_j + \sum_{i=1}^{m} \nu_i = 1,
\end{aligned}$$

$$\mu_j, j = 1, \ldots, p; \quad \nu_i \geqq \varepsilon, i = 1, \ldots, m,$$

where $\alpha > 0$ is a given parameter and $\tilde{d}_k := \max\limits_{\substack{j=1,\ldots,p \\ i=1,\ldots,m}} \left\{ \mu_j(y_{jo} - y_{jk}),\ \nu_i(-x_{io} + x_{ik}) \right\}$.

Remark The formula of $\tilde{d}_k$ means the value of multiplying the maximal component of $(y_{1o} - y_{1k}, \ldots, y_{qo} - y_{qk}, -x_{1o} + x_{1k}, \ldots, -x_{mo} + x_{mk})$ by its corresponding weight. For example, if $(y_{1o} - y_{1k}, -x_{1o} + x_{1k}) = (2, -1)$, then $\tilde{d}_k = 2\mu_1$.

It is noted that when $k = o$, the right-hand side of the inequality constraint in the problem (GDEA) is zero, and hence its optimal value is not greater than zero. The efficiency based on the GDEA model is defined as follows:

Definition 9.2.2 (GDEA efficiency) *For a given positive number α, DMU is said to be GDEA-efficient if and only if the optimal value to the problem (GDEA) is equal to zero. Otherwise, DMU is said to be* GDEA-inefficient.

Table 9.3 shows efficiencies by varying the value of the parameter α in the problem (GDEA). It can be observed that when α is sufficiently small, a GDEA-efficient DMU is FDH-efficient, and if α is sufficiently large, a GDEA-efficient DMU is BCC- or CCR-efficient.

Here, some theoretical properties between GDEA efficiency and DEA efficiencies in the previous subsection are summarized as follows [for detailed proofs of the following theorems, see Yun *et al*. (2004b)]:

Theorem 9.2.3 *A DMU is FDH-efficient if and only if DMU is GDEA-efficient for a sufficiently small $\alpha > 0$.*

Theorem 9.2.4 *A DMU is BCC-efficient if and only if DMU is GDEA-efficient for a sufficiently large $\alpha > 0$.*

Theorem 9.2.5 *If the constraint $\sum_{j=1}^{p} \mu_j y_{jo} = \sum_{i=1}^{m} \nu_i x_{io}$ is added to the problem (GDEA), then a DMU is CCR-efficient if and only if it is GDEA-efficient for a sufficiently large $\alpha > 0$.*

As is stated in the above theorems, various kinds of DEA-efficient frontiers are obtained by changing the value of the parameter α in the GDEA model. Figure 9.3 shows various GDEA-efficient frontiers for several values of α. The DMUs on the

Table 9.3 GDEA efficiencies for the data in Table 9.1.

DMU	*A*	*B*	*C*	*D*	*E*	*F*
$\alpha = 0.001$	0.0	0.0	0.0	−0.002	0.00	−0.502
$\alpha = 0.5$	0.0	0.0	0.0	−1.060	0.00	−1.500
$\alpha = 5$	0.0	0.0	0.0	−7.440	−2.04	−9.600
$\alpha = 10$	0.0	0.0	0.0	−14.420	−4.54	−18.870
$\alpha = 10$ and $\sum_{j=1}^{p} \mu_j y_{jo} = \sum_{i=1}^{m} \nu_i x_{io}$	−4.0	0.0	−14.0	−15.500	−12.25	−25.500

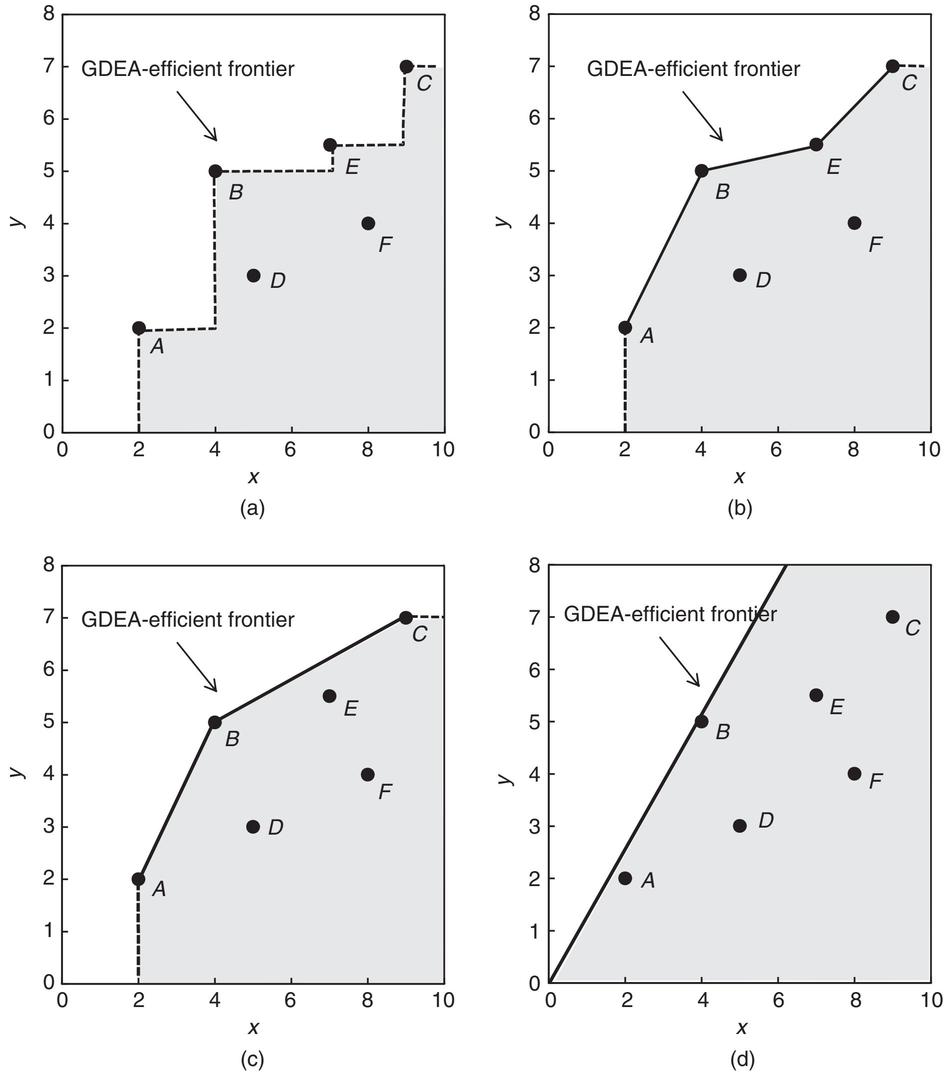

Figure 9.3 GDEA-efficient frontiers for several values of the parameter α (the solid line is the efficient frontier, the dotted line is the weakly efficient frontier and the shadowed area is the production possibility set): (a) $\alpha = 0.001$; (b) $\alpha = 0.5$; (c) $\alpha = 5$; (d) $\alpha = 10$.

GDEA-efficient frontier are GDEA-efficient. It is seen also from Figure 9.3 that as the value of α becomes sufficiently large, the GDEA-efficient frontier changes to a straight line. Varying the value of parameter α, the GDEA model can generate several efficient frontiers, including existing DEA-efficient frontiers such as CCR-, BCC- and FDH-efficient frontiers.

9.3 Generation of Pareto optimal solutions using GDEA and computational intelligence

In this section, several methods combining GDEA and computational intelligence techniques are described for generating good approximate Pareto optimal solutions.

9.3.1 GDEA in fitness evaluation

Recently, multi-objective optimization methods using evolutionary methods, such as GAs, have been studied actively by many researchers (Coello Coello *et al*. 2001; Deb 2001). These approaches are useful for generating Pareto frontiers mainly with two or three objective functions, and decision making can be easily performed on the basis of the visualized Pareto frontier. In generating Pareto frontiers by evolutionary methods, there are two main issues involving the convergence and the diversity of the obtained solutions: (i) how to guide individuals to the real Pareto frontier as close and fast as possible; and (ii) how to keep the diversity of individuals spreading over the whole Pareto frontier at the final generation. The convergence and the diversity of individuals are closely related to fitness evaluation for each individual. In this subsection, we describe several techniques for fitness evaluation in GAs, and introduce the fitness evaluation using GDEA.

GAs were developed by Holland (1975, 1992), and were later applied to function optimization by De Jong (1975). After publication of the book 'Genetic Algorithms in Search, Optimization and Machine Learning' by Goldberg (1989), GAs have attracted considerable attention as a useful optimization tool. Since GAs are a type of population-based search algorithm, they seem to be suited to generate exactly or approximately a set of Pareto optimal solutions.

Basically, GAs are algorithms based on natural selection and natural genetics [more details on GA operators can be found in the book by Deb (2001)]. The natural selection is performed on the basis of the fitness of the individual solutions. Many multi-objective GAs use rank-based methods (ranking method) in the fitness assignment. Originally, the concept of rank was introduced by Goldberg (1989). If an individual is dominated by n_o individuals, then its rank r is given by $n_o + 1$. In Goldberg's ranking method, all non dominated individuals (that is, Pareto optimal solutions in the population) are assigned a rank equal to 1. Then, these individuals are removed from the population and all non dominated individuals in the remaining set of solutions are assigned a rank equal to 2. This procedure is repeated until a rank is assigned to all individuals. On the other hand, the ranking method by Fonseca and Fleming (1993) defines the rank of the solutions as $1+$ the number of dominating individuals in the population.

The value in parentheses for each point in Figure 9.4 shows the rank by these two ranking methods. For generating a good approximate Pareto frontier by population-based methods, points far from the frontier of the population at early stages should be removed in the next generation in order to accelerate the convergence. For example, although the point E has rank 1, it is far from the frontier of the convex hull of the population. At early stages, a rough approximation using the frontier of the convex hull of the population seems to bring faster convergence to the true frontier than approximating non convex parts in detail. At late stages, however, non convex parts should be described in detail

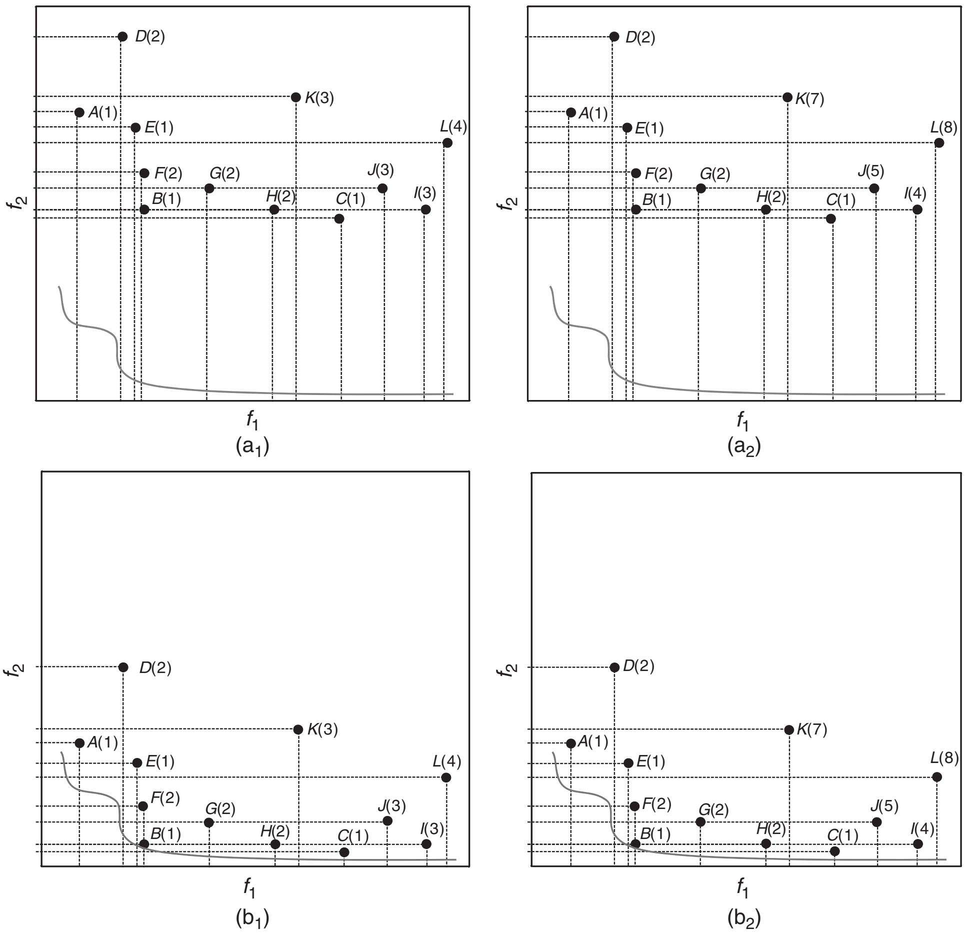

Figure 9.4 Ranking methods [the number in parentheses is the rank of each individual, the curves of (a) and (b) are the true Pareto frontier at an early stage and at late stage, respectively].

for better approximation of the true Pareto frontier. This strategy can be realized by using GDEA. Namely, starting with a relatively large α in GDEA, we can remove the point E at an early stage, because it has a relatively small value of Δ. As the generation proceeds, decreasing the value of α can survive points such as E in a non convex part of the Pareto frontier. A similar discussion can be made for the point D. The point F plays an important role at late stages for deciding the non convex part of the Pareto frontier. Therefore, F should remain in the population at late stages. GDEA can make it. On the other hand, since the rank of I is relatively large, I would be removed from the population by the usual evolutionary method. However, from a viewpoint of diversity, I should remain in the population. GDEA can do this, by taking a large α. The strategy of GDEA is to grasp a rough approximation of the Pareto frontier at early stages, so that the population can be guided to the real Pareto frontier as fast as possible, while at late stages the detailed shape of the Pareto frontier is carved out.

The concept at early stages is the same with the DEA (CCR) method as was suggested by Arakawa *et al*. (1998). DEA gives the fitness of an individual $\mathbf{x}_o$, $o = \mathbf{1}, \ldots, N$ (where N is the population number) as the optimal value to the following problem:

$$\begin{array}{lll} \underset{\theta,\, \boldsymbol{\lambda},\, \mathbf{s}}{\text{minimize}} & \theta - \varepsilon \mathbf{1}^{\top} \mathbf{s} & (\text{DEA}_F) \\ \text{subject to} & \left[\mathbf{f}(\mathbf{x}_1)\ \ldots\ \mathbf{f}(\mathbf{x}_N)\right] \boldsymbol{\lambda} - \theta \mathbf{f}(\mathbf{x}_o) + \mathbf{s} = 0, & \\ & \mathbf{1}^{\top} \lambda = 1, & \\ & \boldsymbol{\lambda} \geqq 0, \mathbf{s} \geqq 0, & \\ & \theta \in \mathbb{R}, \boldsymbol{\lambda} \in \mathbb{R}^N, \quad \mathbf{s} \in \mathbb{R}^m. & \end{array}$$

The optimal value θ to the problem (DEA_F) indicates the relative closeness of $\mathbf{f}(\mathbf{x}_o)$ to the DEA-efficient frontier, which can be regarded as a roughly approximate Pareto frontier based on the given population. As shown in Figure 9.5, the fitness of an individual D is the ratio of OD' to OD, and the fitness of an individual K is the ratio of OK' to OK. Thus, using DEA, individuals such as E and G who are far from the DEA-efficient frontier, have low fitness, and hence can be removed at relatively early stages. However, since the DEA-efficient frontier is always convex, this DEA method may not generate the solutions in the sunken part of the Pareto frontier, because individuals such as E, F and H are removed from the population at late stages.

Taking the advantages and overcoming the shortcomings of ranking methods and the DEA method, Yun *et al*. (2001) suggested a fitness evaluation using the GDEA model

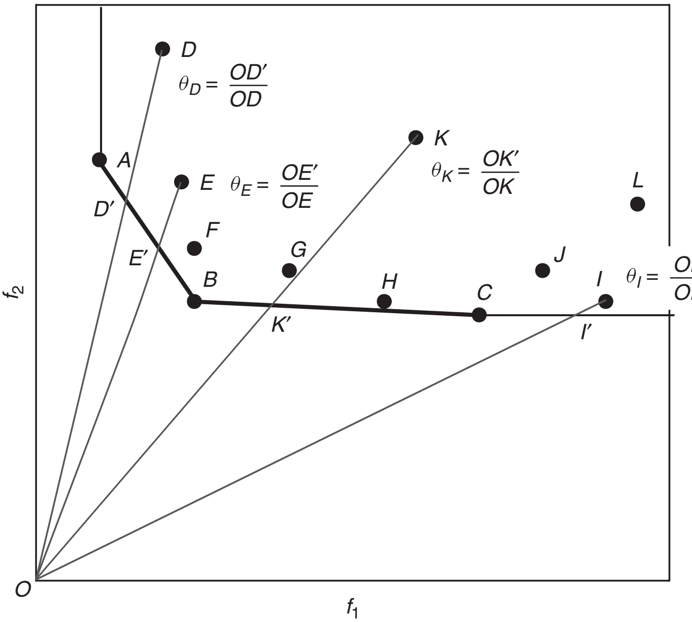

Figure 9.5 Fitness evaluation by the DEA method (the solid line is the CCR-efficient frontier).

as follows:

$$\begin{aligned}
&\underset{\Delta,\nu}{\text{maximize}} && \Delta && \text{(GDEA}_F\text{)}\\
&\text{subject to} && \Delta \leqq \tilde{d}_k - \alpha \sum_{i=1}^{m} \nu_i \left(f_i(\mathbf{x}_o) - f_i(\mathbf{x}_k) \right), \quad k = 1, \ldots, N, &&\\
& && \sum_{i=1}^{m} \nu_i = 1, &&\\
& && \nu_i \geqq \varepsilon, i = 1, \ldots, m, &&
\end{aligned}$$

where $\tilde{d}_k = \max\limits_{i=1,\ldots,m} \left\{ \nu_i \left(-f_i(\mathbf{x}_o) + f_i(\mathbf{x}_k) \right) \right\}$ and α is a value that decreases monotonically as the generation proceeds. For example, α can be given by

$$\alpha(t) := \omega \cdot \exp\ (-\beta \cdot t), t = 0, 1, \ldots, T,$$

where ω, β and T are positive fixed numbers, $\omega (= \alpha(0))$ is determined to be sufficiently large (e.g., 10, 10^2, 10^3), and T is the generation number. For a given ω and T, β is decided by solving the equation $\alpha(T) = \omega \cdot \exp\ (-\beta \cdot T) \fallingdotseq 0$.

The degree of GDEA efficiency for an individual $\mathbf{x}_o$ is given by the optimal value Δ^* to the problem (GDEA$_F$), and is used in fitness evaluation. Therefore, the selection of an individual to survive in the subsequent generations is determined by the degree of GDEA efficiency, i.e., if Δ^* equals zero, the individual remains in the next generation. Taking advantage of the properties of GDEA, it is possible to keep the strong points of ranking methods and DEA, and at the same time, to overcome the shortcomings of existing methods. In particular, using a large α can remove individuals located far from the GDEA-efficient frontier, whereas a small α can generate the non convex part of the Pareto frontier (Figure 9.6).

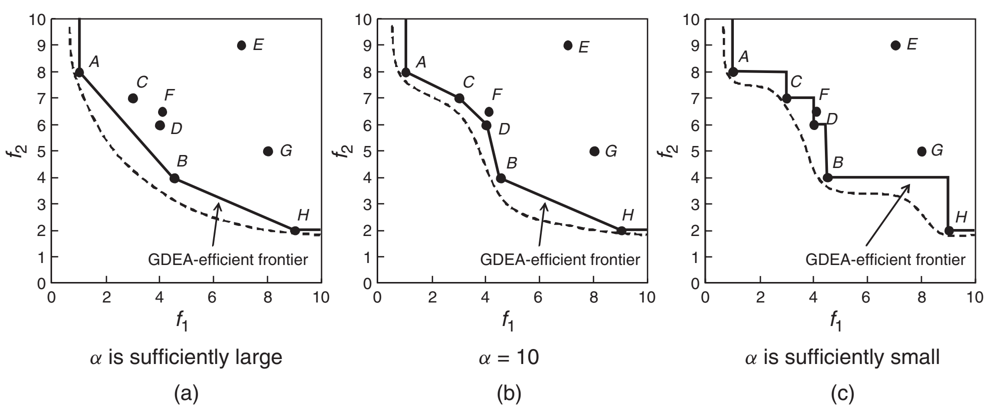

Figure 9.6 Geometric interpretation of α in (GDEA$_F$) (points are individuals, the solid line is the GDEA-efficient frontier, and the dotted line is an approximate Pareto frontier): (a) initial generation; (b) intermediate generation; (c) final generation.

Furthermore, Yun *et al*. (2004a) proposed the method of combining an aspiration level approach with GDEA, in order to find not the whole Pareto frontier but the most interesting part of it for decision makers. As stated earlier, the fitness in the GDEA method is evaluated by the optimal value Δ^* to the problem (GDEA$_F$), which represents a relative degree of how close $\mathbf{f}(\mathbf{x}_o)$ is to the GDEA-efficient frontier.

Given the aspiration level of the decision maker $\bar{\mathbf{f}} = \left(\overline{f}_1, \ldots, \overline{f}_m\right)^\top$ and an ideal point $\mathbf{f}^* = \left(f_1^*, \ldots, f_m^*\right)^\top$, the fitness for an individual $\mathbf{x}_o$ is evaluated by solving the problem (ALGDEA$_F$):

$$\begin{array}{lll} \underset{\Delta,\,\nu}{\text{maximize}} & \Delta - \rho H\left(\mathbf{W}\left(\mathbf{f}(\mathbf{x}_o) - \bar{\mathbf{f}}\right)\right) & (\text{ALGDEA}_F) \\ \text{subject to} & \Delta \leqq \tilde{d}_k - \alpha \displaystyle\sum_{i=1}^{m} \nu_i (f_i(\mathbf{x}_o) - f_i(\mathbf{x}_k)),\ k = 1, \ldots, N, & \\ & \displaystyle\sum_{i=1}^{m} \nu_i = 1, & \\ & \nu_i \geqq \varepsilon,\ i = 1, \ldots, m, & \end{array}$$

where $H(\mathbf{y}) = \max\{y_1, \ldots, y_m\}$, $\mathbf{y} = (y_1, \ldots, y_m)^\top$, $\mathbf{W}$ is the diagonal matrix with its elements defined as $w_i = 1/(\overline{f}_i - f_i^*)$, $i = 1, \ldots, m$, and ρ is an appropriately given positive constant. For instance, ρ is taken as a large number when we search as close Pareto optimal values to the given aspiration level as possible.

Consequently, the optimal value to the problem (ALGDEA$_F$) represents the degree of how close $\mathbf{f}(\mathbf{x}_o)$ is to the Pareto frontier and to the given aspiration level. For example, as shown in Figure 9.7, let $\bar{\mathbf{f}}$ be a given aspiration level and $\mathbf{f}^*$ be an ideal point. Then, in the fitness assignment by (ALGDEA$_F$), individuals in the neighborhood of the Pareto

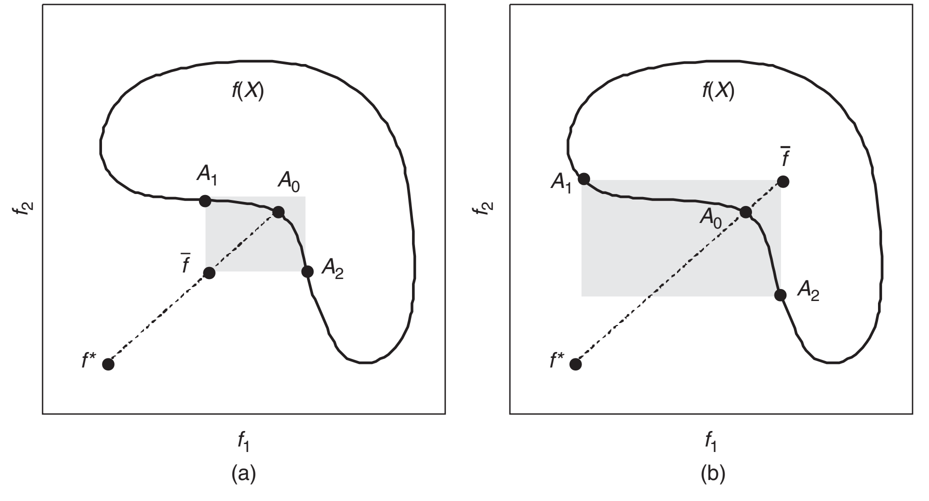

Figure 9.7 Geometric interpretation of the problem (ALGDEA$_F$): (a) $\bar{\mathbf{f}}$ is infeasible; (b) $\bar{\mathbf{f}}$ is feasible.

optimal value $A0$ have good fitness, and the Pareto optimal region shadowed in Figure 9.7 is considered to be the most interesting part for the decision maker.

From approximate Pareto optimal solutions generated in the region, representative alternatives are selected using the following measures:

$$A0 = \mathbf{f}(\hat{\mathbf{x}}^0), \text{where } \hat{\mathbf{x}}^0 = \arg\min_{\mathbf{x}_j} \max_{i=1,\ldots,m} w_i \left(f_i(\mathbf{x}_j) - \bar{f}_i \right),$$

$$Ai = \mathbf{f}(\hat{\mathbf{x}}^i), \text{where } \hat{\mathbf{x}}^i = \arg\min_{\mathbf{x}_j} w_i \left(f_i(\mathbf{x}_j) - \bar{f}_i \right), i = 1, \ldots, m,$$

where $\mathbf{x}_j$ is in the set of generated Pareto optimal solutions. Note that $\hat{\mathbf{x}}^0$ can be approximately given as the one which gives the maximum among optimal values to the problem (ALGDEA$_F$) for each $\mathbf{x}_j$.

9.3.2 GDEA in deciding the parameters of multi-objective PSO

Next, we give an overview of utilizing GDEA in deciding adaptively the parameters in PSO for solving multi-objective optimization problems. PSO was proposed by Kennedy and Eberhart (2001) based on the simulation of the social behavior in bird flocks in order to find their food. Here, a particle is an individual in a swarm, which corresponds to a population in a GA. Each particle in PSO has its position and the velocity in the search space, and moves towards the goal (optimal solution). The position of each particle is changed according to its own experience and the information from other particles.

In a single-objective optimization problem, let $\mathbf{x}_i^t$ be the position and $\mathbf{v}_i^t$ the velocity at the time step t for a particle p_i. Then, the next position $\mathbf{x}_i^{t+1}$ is updated by the following (Figure 9.8):

$$\begin{aligned} \mathbf{x}_i^{t+1} &= \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \\ \mathbf{v}_i^{t+1} &= w\mathbf{v}_i^t + c_1 r_1 (\mathbf{x}_{pb_i} - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{x}_{gb} - \mathbf{x}_i^t), \end{aligned} \tag{9.2}$$

where w is the inertia weight, c_1, c_2 are positive acceleration components, and r_1, $r_2 \in [0, 1]$ are uniformly distributed random values for considering a stochastic element in the search. The notations $\mathbf{x}_{pb_i}$ and $\mathbf{x}_{gb}$ are usually given by the personal best position of the particle p_i and the best position of the entire swarm, respectively.

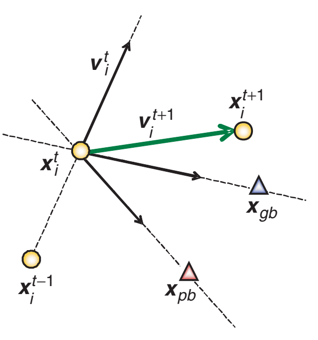

Figure 9.8 Geometric interpretation of the updating velocity in PSO.

However, when applying PSO to multi-objective optimization, the concept of 'best (optimal)' cannot be defined as a unique one. Consequently, it is not easy to decide $\mathbf{x}_{pb_i}$ and $\mathbf{x}_{gb}$ which are important for updating the velocity (9.2). Up to now, several PSO algorithms have been developed for multi-objective optimization (Alvarez-Benitez *et al*. 2005; Li 2003; Moore and Chapman 1999; Mostaghim and Teich 2003; Nebro *et al*. 2009; Sierra and Coello Coello 2005; Toscano-Pulido and Coello Coello 2004; Toscano-Pulido *et al*. 2007). Many other proposals can be referred to in Reyes-Sierra and Coello Coello (2006), and experimental comparisons for the state-of-the-art multi-objective PSOs are reported in Durillo *et al*. (2009).

GDEA is available for the update of the velocity (9.2), and the idea of deciding the parameters in a multi-objective PSO is introduced in the following.

Given the positive number α, the dual problem to (GDEA$_F$) described in Section 9.3.1 is formulated as follows:

$$\begin{array}{lll} \underset{\delta,\boldsymbol{\lambda},\mathbf{s}}{\text{minimize}} & \delta - \varepsilon \mathbf{1}^\top \mathbf{s} & (\text{GDEA}_P) \\ \text{subject to} & \left\{\alpha\left(-F_o + F\right) + D\right\}\boldsymbol{\lambda} - \delta \mathbf{1} + \mathbf{s} = 0, & \\ & \mathbf{1}^\top \lambda = 1, & \\ & \boldsymbol{\lambda} \geqq \mathbf{0}, \mathbf{s} \geqq 0, & \\ & \boldsymbol{\lambda} \in \mathbb{R}^N, \mathbf{s} \in \mathbb{R}^m, & \end{array}$$

where

$$F = \left[\mathbf{f}(\mathbf{x}_1)\ \cdots\ \mathbf{f}(\mathbf{x}_N)\right] \in \mathbb{R}^m \times \mathbb{R}^N,$$

$$F_o = \left[\mathbf{f}(\mathbf{x}_o)\ \cdots\ \mathbf{f}(\mathbf{x}_o)\right] \in \mathbb{R}^m \times \mathbb{R}^N,$$

D : matrix replaced by 0 except for the maximal component in each column of the matrix $\left(-F_o + F\right)$

As explained in Section 9.2, GDEA evaluates the relative proximity of $\mathbf{f}(\mathbf{x}_o)$ to the GDEA-efficient frontier generated by the given data set. The optimal solution δ^* represents the degree of the proximity to the GDEA-efficient frontier in the objective function space. The particle with $\delta^* = 0$ is GDEA-efficient, otherwise the particle is GDEA-inefficient. In the latter case, the absolute value $|\delta^*| > 0$ is the degree of GDEA inefficiency. For example, Table 9.4 shows the results by GDEA for sufficiently large $\alpha (= 10$ in this case). As seen from the results, particles p_{A}, p_{B} and p_{H} on the GDEA-efficient frontier have the optimal solution $\delta^* = 0$. On the other hand, particles with a negative δ^* are away from the GDEA-efficient frontier. Although both particles p_{D} and p_{G} are GDEA-inefficient, particle p_{G} is more GDEA-inefficient than particle p_{D} because $|\delta^*_{\mathrm{G}}| = 65.75 > |\delta^*_{\mathrm{D}}| = 6.96$. Also, this means that $\mathbf{f}(\mathbf{x}_{\mathrm{G}})$ is relatively farther away from the GDEA-efficient frontier compared with $\mathbf{f}(\mathbf{x}_{\mathrm{D}})$.

It can be thought that GDEA-efficient particles should keep the current direction, while GDEA-inefficient particles should change the moving direction. Therefore, it is natural that the inertia parameter w for each particle in the velocity (9.2) is taken according to the value of δ in order to improve the convergence of the search. That is, it is appropriate to take large w for GDEA-efficient particles and small w for GDEA-inefficient particles.

Table 9.4 Results by GDEA for $\alpha = 10$.

i	$(f_1,\ f_2)$	δ_i^*	$\boldsymbol{\lambda}^* = (\lambda_A^*, \ldots, \lambda_J^*)^\top$
A	(1, 12)	0.00	$\lambda_A^* = 1$
B	(5, 5)	0.00	$\lambda_B^* = 1$
C	(3, 11)	−8.48	$\lambda_A^* = 0.73, \lambda_B^* = 0.27$
D	(5, 7)	−6.96	$\lambda_A^* = 0.17, \lambda_B^* = 0.83$
E	(8, 14)	−53.09	$\lambda_A^* = 0.54, \lambda_B^* = 0.46$
F	(4, 8)	−3.62	$\lambda_A^* = 0.36, \lambda_B^* = 0.64$
G	(14, 10)	−65.75	$\lambda_B^* = 0.69, \lambda_H^* = 0.31$
H	(13, 1)	0.00	$\lambda_H^* = 1$
I	(9, 4)	−5.73	$\lambda_B^* = 0.59, \lambda_H^* = 0.41$
J	(11, 3)	−5.77	$\lambda_B^* = 0.34, \quad \lambda_H^* = 0.66$

The inertia parameter w_i for a particle p_i can be determined based on the normalized GDEA-efficient value δ_i^* as follows:

$$w_i = \frac{\delta_i^* - \min\limits_{j=1,\ldots,N} \delta_j}{\max\limits_{j=1,\ldots,N} \delta_j^* - \min\limits_{j=1,\ldots,N} \delta_j^*} = 1 - \frac{\delta_i^*}{\min\limits_{j=1,\ldots,N} \delta_j^*}, \qquad i = 1, \ldots, N. \tag{9.3}$$

The last half part of (9.3) is derived from $\max\limits_{j=1,\ldots,N} \delta_j^* = 0$.

Moreover, the optimal solution $\boldsymbol{\lambda}^*$ to the dual problem (GDEA$_P$) provides information on reference points. For a particle p_i, let

$$I_i = \left\{ j \mid \lambda_j^* > 0,\ j = 1, \ldots, N \right\}.$$

Then, the GDEA-inefficient particle p_i is dominated by the GDEA-efficient $p_j,\ j \in I_i$ in the sense of GDEA efficiency. Therefore, the set I_i is regarded as the information on good (efficient) neighbor particles for a particle p_i. In order to keep the diversity of solutions, utilizing the optimal solution $\boldsymbol{\lambda}^*$, we give the second term and the third term of the velocity (9.2) in the following form:

$$c_1 r_1 (\mathbf{x}_{pb_i} - \mathbf{x}_i) + c_2 r_2 (\mathbf{x}_{gb} - \mathbf{x}_i) := \sum_{j=1}^{N} \lambda_j^* r_j (\mathbf{x}_j - \mathbf{x}_i) = \sum_{j \in I_i} \lambda_j^* r_j (\mathbf{x}_j - \mathbf{x}_i), \tag{9.4}$$

where $r_j \in [0, 1]$ is a random value.

For example, the form (9.4) for the particle p_{C} becomes $0.73 r_1 (\mathbf{x}_{\mathrm{A}} - \mathbf{x}_{\mathrm{C}}) + 0.27 r_2 (\mathbf{x}_{\mathrm{B}} - \mathbf{x}_{\mathrm{C}})$. As shown in Table 9.4, GDEA-inefficient particles $p_i,\ i =$ C, D, E, F have the indices set $I_i = \{\mathrm{A, B}\}$, and based on the positions $\mathbf{f}(\mathbf{x}_{\mathrm{A}})$ and $\mathbf{f}(\mathbf{x}_{\mathrm{B}})$ in the objective function space, the particles move toward the neighborhood of the line AB in Figure 9.9. Similarly, particles $p_i,\ i =$ G, I, J have the indices set $I_i = \{\mathrm{B, H}\}$, and they go toward the neighborhood of the line BH in Figure 9.9 on the basis of $\mathbf{f}(\mathbf{x}_{\mathrm{B}})$ and $\mathbf{f}(\mathbf{x}_{\mathrm{H}})$. Using the information of $\boldsymbol{\lambda}$ yields an effect on clustering particles, and consequently, it is expected to keep the diversity of solutions in the search.

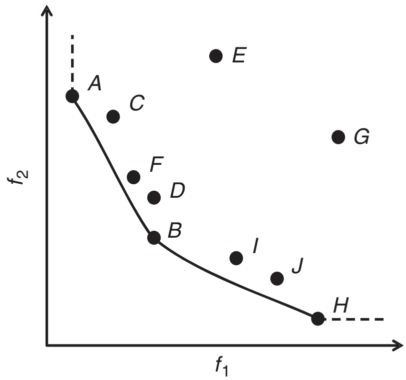

Figure 9.9 GDEA-efficient frontier for Table 9.4 (the points are the individuals, the solid line is the GDEA-efficient frontier and the dotted line is the weakly efficient frontier).

9.3.3 Expected improvement for multi-objective optimization using GDEA

As explained in Section 9.3.1, GDEA can measure a relative inefficiency for each individual, while we cannot know how efficient an individual is. To this end, we formulate the extended GDEA model:

$$\begin{aligned} \underset{\delta,\boldsymbol{\lambda},\mathbf{s}}{\text{minimize}} \quad & \delta - \varepsilon \mathbf{1}^\top \mathbf{s} && (\text{GDEA}_E) \\ \text{subject to} \quad & \left\{\alpha\left(-F_o + F\right) + D\right\}\boldsymbol{\lambda} - \delta\mathbf{1} + \mathbf{s} = 0, \\ & 1^\top \lambda = \mathbf{1}, \\ & \boldsymbol{\lambda} \geqq \mathbf{0}, \mathbf{s} \geqq 0, \\ & \boldsymbol{\lambda} \in \mathbb{R}^{N-1}, \mathbf{s} \in \mathbb{R}^m, \end{aligned}$$

where $F = \left[\mathbf{f}(\mathbf{x}_1) \ \cdots \ \mathbf{f}(\mathbf{x}_{o-1})\ \mathbf{f}(\mathbf{x}_{o+1}) \ \cdots \ \mathbf{f}(\mathbf{x}_N)\right]$ and $F_o = \left[\mathbf{f}(\mathbf{x}_o) \ \cdots \ \mathbf{f}(\mathbf{x}_o)\right] \in \mathbb{R}^m \times \mathbb{R}^{N-1}$.

The above model GDEA$_E$ evaluates a relative distance from the GDEA-efficient frontier which is generated by the data set $\{\mathbf{f}(\mathbf{x}_1), \ldots, \mathbf{f}(\mathbf{x}_N)\}\backslash\mathbf{f}(\mathbf{x}_o)$.

As described in the previous section, GDEA-inefficient individuals do not contribute to the generation of the GDEA-efficient frontier. Conversely, the GDEA-efficient frontier is determined only by GDEA-efficient individuals, and thus may be altered by excluding an individual from the population. Figure 9.10 shows the GDEA-efficient frontier ($\alpha = 10$) excluding individual p_{B} from the data of Table 9.4. For GDEA-efficient individuals p_{A}, p_{B} and p_{H}, their efficient values by the problem (GDEA$_E$) change to:

$$\delta^*_{\mathrm{A}} = 22.00, \quad \delta^*_{\mathrm{B}} = 12.74, \quad \delta^*_{\mathrm{H}} = 22.00.$$

The above efficient values indicate how far $\mathbf{f}(\mathbf{x}_o)$ is from the GDEA-efficient frontier generated without the individual, and it can be regarded that individuals with larger values of δ are more evolved for generating a better approximation of the Pareto frontier.

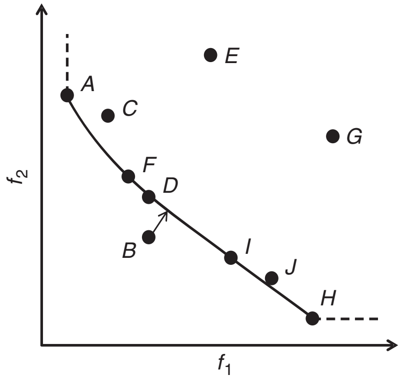

Figure 9.10 Extended GDEA-efficient frontier for Table 9.4 ($\alpha = 10$; the points are the individuals, the solid line is the GDEA-efficient frontier generated by the data except for B and the dotted line is the weakly efficient frontier).

Therefore, the value of GDEA efficiency by model GDEA$_E$ can constitute a measure of how GDEA-efficient individuals evolve with respect to the current GDEA-efficient frontier. Consequently, keeping such individuals with larger values in the population, it is expected to provide good offspring in the next generation.

In this subsection, we describe a method utilizing GDEA$_E$ to make good offspring (repopulation at the next generation from the current parent population) in a real-coded GA.

In many practical engineering design problems, the form of objective functions cannot be given explicitly in terms of design variables. In this case, the values of the objective functions for some given values of the design variables, are obtained by some analysis/experiments such as structural analysis, fluid mechanic analysis, thermodynamic analysis and so on. Usually, these analyses are considerably time consuming to perform. Therefore, in such real application problems, it is important to make the number of function evaluations as small as possible for finding an optimal solution to the predicted objective function based on the sampled data set. For that purpose, EI was suggested as a criterion for selecting new sample points in efficient global optimization (EGO) for single-objective optimization problems (Jones 2001; Jones *et al*. 1998; Schonlau 1997; Schonlau *et al*. 1998).

Let $(\mathbf{x}_i, y_i), \quad i = 1, \ldots, N$ be sampled data, where $\mathbf{x}_i$ is a vector of design variables, and y_i is the value of the objective function at point $\mathbf{x}_i$. Further, let $\hat{y}(\mathbf{x})$ be a predicted objective function based on the given sampled data. Suppose that for a point $\mathbf{x}$, Y is a random variable such that $Y \sim N(\hat{y}, \hat{s}^2)$. In the Kriging model-based EGO, the predicted function $\hat{y}$ and the mean squared error of the prediction $\hat{s}$ are estimated as follows [for details, see Jones *et al*. (1998)]:

$$\begin{aligned}
\hat{y}(\mathbf{x}) &= \hat{\mu} + \mathbf{r}_{\mathbf{x}}^{\top}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \\
\hat{s}^2(\mathbf{x}) &= \hat{\sigma}^2\left(1 - \mathbf{r}_{\mathbf{x}}^{\top}\mathbf{R}^{-1}\mathbf{r}_{\mathbf{x}} + \frac{(1 - \mathbf{1}^{\top}\mathbf{R}^{-1}\mathbf{r}_{\mathbf{x}})^2}{\mathbf{1}^{\top}\mathbf{R}^{-1}\mathbf{1}}\right),
\end{aligned} \tag{9.5}$$

where

$$\hat{\mu} = \left(\mathbf{1}^\top \mathbf{R}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^\top \mathbf{R}^{-1} \mathbf{y}, \quad \hat{\sigma}^2 = \frac{1}{N} (\mathbf{y} - \mathbf{1}\hat{\mu})^\top \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}),$$

$$\mathbf{R} = \left(\mathbf{r}_{\mathbf{x}_1}, \cdots, \mathbf{r}_{\mathbf{x}_N}\right), \quad \mathbf{r}_{\mathbf{x}} = \left(R(\mathbf{x}_1, \mathbf{x}), \cdots, R(\mathbf{x}_N, \mathbf{x})\right)^\top,$$

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\mathrm{r}^2}\right) : \text{the correlation function}^2\text{, where r is a given parameter,}$$

$$\mathbf{y} = (y_1, \ldots, y_N)^\top, \quad \mathbf{1} = (1, \ldots, 1)^\top.$$

Then, the improvement I over the current best value $f_{\min} = \min\{y_1, \ldots, y_N\}$ is defined by:

$$I(\mathbf{x}) = \max\left(f_{\min} - Y,\ 0\right)$$

and the expected value of improvement (EI) can be computed as:

$$E[I(\mathbf{x})] = \begin{cases} \left(f_{\min} - \hat{y}(\mathbf{x})\right) \Phi\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x}) \phi\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) & \text{if} \quad \hat{s}(\mathbf{x}) > 0 \\ 0 \quad \text{if} \quad \hat{s}(\mathbf{x}) = 0 \end{cases}, \tag{9.6}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the density function and the cumulative distribution function of the standard normal distribution, respectively.

The criterion is to maximize the expected improvement (9.6) which can be interpreted from two viewpoints: (a) finding a good approximate optimal solution (local search/information); and (b) predicting unknown objective function values (global search/information). Differentiating Equation (9.6) with respect to $\hat{y}$ and $\hat{s}$, shows that EI is decreasing with respect to $\hat{y}$ and increasing with respect to $\hat{s}$. In other words, EI is larger for smaller $\hat{y}$ and for larger $\hat{s}$. Note that $\hat{y}$ is a predicted objective function value and $\hat{s}$ is the standard error of the prediction, which represents the uncertainty for the prediction. From Equation (9.5), it can be shown that $\hat{s} = 0$ holds for the existing observed data, that is, there is no error at the given sample points, while the value of $\hat{s}$ is large in the region nearby the area where there are a smaller number of sampled data. Thus, additional points should be selected from this region by maximizing EI in order to obtain a more accurate prediction of the objective function value and to improve the optimal solution.

In addition, for searching more globally or locally according to situations, EI was generalized: the improvement I^g for a non-negative parameter g is defined by:

$$I^g(\mathbf{x}) = \max\left\{\left(f_{\min} - Y\right)^g, 0\right\},$$

and the *generalized expected improvement* (Schonlau 1997) is expressed by:

$$\mathrm{E}[I^g(\mathbf{x})] = \hat{s}^g \sum_{i=0}^{g} (-1)^i \left(\frac{g!}{i!(g-i)!}\right) \left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)^{g-i} T_i, \tag{9.7}$$

[2] The correlation function R indicates the correlation between the errors at two points $\mathbf{x}_i$ and $\mathbf{x}_j$.

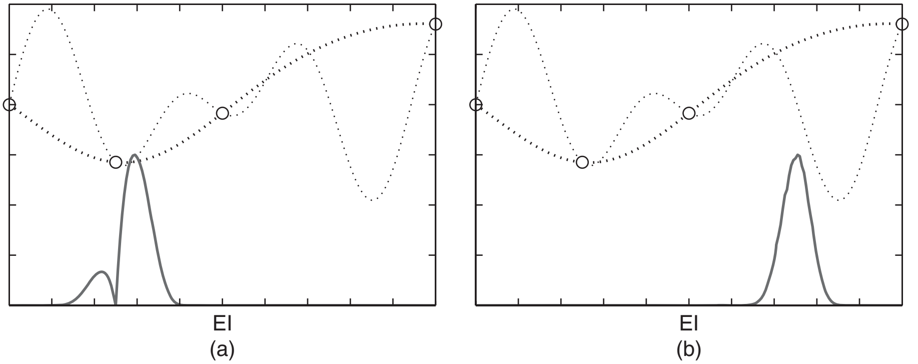

Figure 9.11 Expected improvements (solid line) (the dotted line is the true function, the dashed line is the predicted function $\hat{y}$, and the circles are sample points); (a) $g = 1$; (b) $g = 10$.

where

$$T_0 = \Phi\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right),$$

$$T_1 = -\phi\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right),$$

$$T_i = -\phi\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)\left(\frac{f_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right)^{i-1} + (i-1)T_{i-2}, i = 2, 3, \ldots .$$

Figure 9.11 shows the EI for the cases of $g = 1, 10$. As shown in Figure 9.11(a) for $g = 1$, the EI is larger around the current best point to the predicted function, while for a sufficiently large $g (= 10$ in this example) of Figure 9.11(b), the EI has a large value in the sparse area of the sampled points.

For extending the EI to the cases of multi-objective optimization, the GDEA-efficient value is used as the predicted function. The EI is calculated based on GDEA-efficient values δ_i for individuals $\mathbf{x}_i$, $i = 1, \ldots, N$, and the next sample points (recombination/repopulation) are selected by maximizing EI. That is, a method combining GDEA and EI makes a recombination in such a way that the GDEA-efficient value of the fitness value in GA is improved (Figure 9.12). For the given sampled data $(\mathbf{x}_i, \delta_i), i = 1, \ldots, N$:

(i) Choose one pair (●) randomly from the current population: $\mathbf{x}^{(1)} = (x_1^{(1)}, \ldots, x_n^{(1)})^\top$, $\mathbf{x}^{(2)} = (x_1^{(2)}, \ldots, x_n^{(2)})^\top$.

(ii) Generate N_c points (□ and ■) by using the blend crossover (Eshelman and Schaffer 1993): for the given parameter $a > 0$,

$$d_i = |x_i^{(1)} - x_i^{(2)}|, i = 1, \ldots, n$$

$$x_i^L = \min\left(x_i^{(1)}, x_i^{(2)}\right) - ad_i, i = 1, \ldots, n$$

$$x_i^U = \max\left(x_i^{(1)}, x_i^{(2)}\right) + ad_i, i = 1, \ldots, n.$$

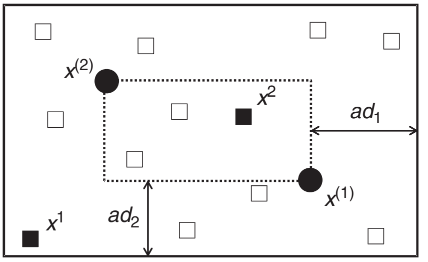

Figure 9.12 Scheme of offspring generated by using GDEA and EI.

(iii) Calculate the value of EI (9.7) for candidate points and choose the point (■) with maximal EI:

$$\mathbf{x}^1 = \arg \max_{j=1,\dots N_c} E[I(\mathbf{x}^j)]$$

$$\mathbf{x}^2 = \arg \max_{j=1,\dots N_c} E[I^{10}(\mathbf{x}^j)].$$

Here, we show the results for the KUR problem (Kursawe, 1991) by the method combining EI and GDEA comparing with NSGA-II (Deb et al., 2002), SPEA2 (Zitzler *et al*., 2001), and OMOPSO (Sierra and Coello Coello, 2005), which are well-known evolutionary methods.

$$\begin{aligned}
&\underset{\mathbf{x}}{\text{minimize}} && f_1 := \sum_{i=1}^{2} -10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right) \\
&\underset{\mathbf{x}}{\text{minimize}} && f_2 := \sum_{i=1}^{3} \left[|x_i|^{0.8} + 5\sin(x_i^3)\right] \\
&\text{subject to} && -5 \leqq x_i \leqq 5, i = 1, 2, 3.
\end{aligned}$$

Figure 9.13–9.16 show the solutions obtained by each method at iteration times $t = 10, 20, 30$, respectively. As seen from the results, the solutions obtained using GDEA converge when $t = 20$, and GDEA provides good approximate solutions compared with the conventional ranking-based methods.

9.4 Summary

In this chapter, we have discussed several methods utilizing the GDEA model in MCDM. GDEA can evaluate efficiencies of several DEA models in a unified way. It has been observed that, taking those advantages of GDEA enables the use of the model in meta-heuristic methods in order to effectively generate Pareto optimal solutions for multi-objective (in particular, two objectives) decision making problems. Combining GDEA

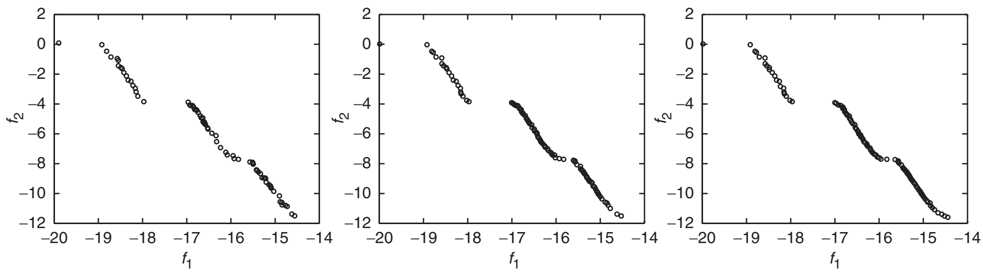

Figure 9.13 Solutions by the method combining GDEA and EI: (a) $t = 10$; (b) $t = 20$; (c) $t = 30$.

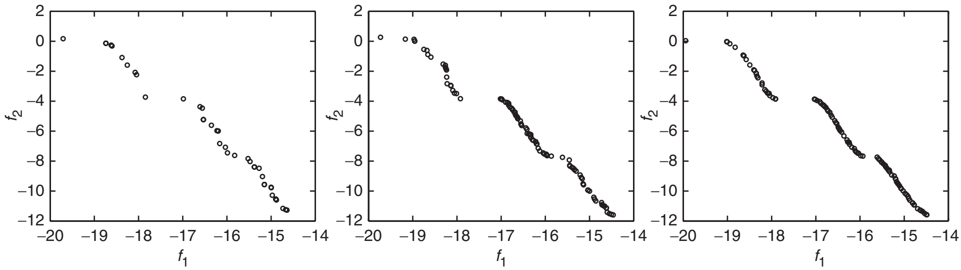

Figure 9.14 Solutions by NSGA-II: (a) $t = 10$; (b) $t = 20$; (c) $t = 30$.

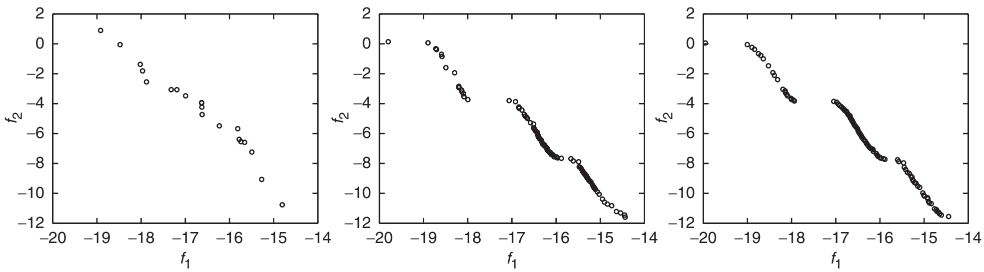

Figure 9.15 Solutions by SPEA2: (a) $t = 10$; (b) $t = 20$; (c) $t = 30$.

and the aspiration level method makes it possible to show the decision makers' most interesting part of Pareto optimal solutions, that is, the solutions closer to their aspiration level. Particularly, we have given an overview of the GDEA method in deciding adaptively the parameters in PSO for solving multi-objective optimization problems, and finally, we have introduced a method combining GDEA and EI for making good offspring in a real-coded GA.

The main benefits of the GDEA model are that it can measure a relative distance for each alternative to the so-called GDEA-efficient frontier and give information on

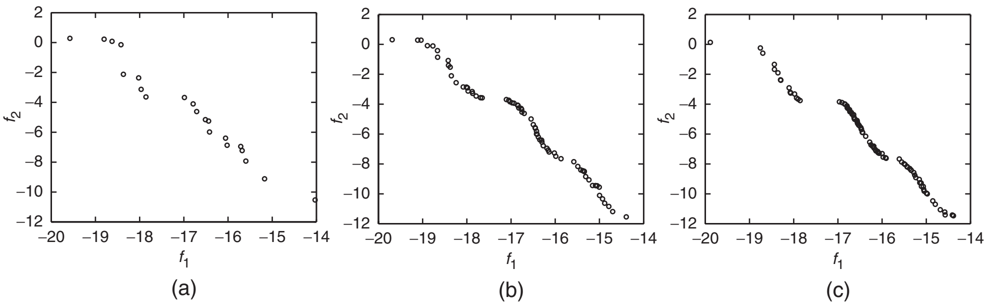

Figure 9.16 Solutions by OMOPSO: (a) $t = 10$; (b) $t = 20$; (c) $t = 30$.

reference points by which the alternative is dominated. From this information, one may see how efficient or inefficient each alternative is, and also how and what should be done to make inefficient alternatives efficient. This is important in supporting decision making with multiple criteria. Therefore, the GDEA model can be regarded as helpful in MCDM, and also applicable to a wide range of real problems.

References

Agrell P and Tind J (1998) An extension of the DEA-MCDM liaison for the free disposable hull model. Technical Report 3, Department of Operations Research, University of Copenhagen, Copenhagen.

Alvarez-Benitez JE, Everson RM and Fieldsend JE (2005) A MOPSO algorithm based exclusively on Pareto dominance concepts. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization*, *EMO'05*. Springer-Verlag, Berlin, pp. 459–473.

Arakawa M, Nakayama H, Hagiwara I and Yamakawa H (1998) Multiobjective optimization using adaptive range genetic algorithms with data envelopment analysis. *A Collection of Technical Papers on 7th Symposium on Multidisciplinary Analysis and Optimization (TP98-4970), AIAA*, vol. 3, pp. 2074–2082.

Banker RD, Charnes A and Cooper WW (1984) Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science* **30**, 1078–1092.

Charnes A, Cooper WW and Rhodes E (1978) Measuring the efficiency of decision making units. *European Journal of Operational Research* **2**, 429–444.

Charnes A, Cooper WW and Rhodes E (1979) Short communication: Measuring the efficiency of decision making units. *European Journal of Operational Research* **3**, 339.

Coello Coello CA, Van Veldhuizen DA and Lamont GB (2001) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Dordrecht.

Cooper WW, Seiford LM and Tone K (2007) *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. Springer, New York.

De Jong KA (1975) *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan.

Deb K (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd, Chichester.

Deb K, Pratap A, Agarwal S and Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Deprins D, Simar L and Tulkens H (1984) Measuring labor-efficiency in post offices. In *The Performance of Public Enterprises: Concepts and Measurements* (eds Marchand M, Pestieu P and Tulkens H). North Holland, Amsterdam, pp. 247–263.

Durillo JJ, García-Nieto J, Nebro AJ, Coello Coello CA, Luna F and Alba E (2009) Multi-objective particle swarm optimizers: An experimental comparison. In *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, *EMO'09*. Springer-Verlag, Berlin, pp. 495–509.

Eshelman LJ and Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. In *Proceedings of the 2nd Workshop on Foundations of Genetic Algorithms* (ed. Whitley LD). Morgan Kaufmann, San Mateo, CA, pp. 187–202.

Fonseca CM and Fleming PJ (1993) Genetic algorithm for multiobjective optimization, formulation, discussion and generalization. In *Proceedings of the 5th International Conference: Genetic Algorithms* (ed. Forrest S). Morgan Kaufmann, San Mateo, pp. 416–423.

Goldberg DE (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Boston, MA.

Holland JH (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Holland JH (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, Cambridge, MA.

Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization* **19**, 345–383.

Jones DR, Schonlau M and Welch WJ (1998) Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13**, 455–492.

Kennedy J and Eberhart RC (2001) *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA.

Kursawe F (1991) A variant of evolution strategies for vector optimization. In *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, *PPSN I*. Springer-Verlag, London, pp. 193–197.

Li X (2003) A non-dominated sorting particle swarm optimizer for multiobjective optimization *Genetic and Evolutionary Computation – GECCO 2003*, vol. 2723 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 37–48.

Moore J and Chapman R (1999) Application of particle swarm to multiobjective optimization. Technical Report, Auburn University, Alabama.

Mostaghim S and Teich J (2003) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *2003 IEEE Swarm Intelligence Symposium Proceedings*. IEEE Press, New York, NY, pp. 26–33.

Nebro AJ, Durillo JJ, García-Nieto J, Coello Coello CA, Luna F and Alba E (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization. *2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)*. IEEE Press, New York, pp. 66–73.

Pareto V (1906) *Manuale Di Economia Politica (Manual of Political Economy)*. Macmillan, London. Translated by Schwier AS.

Reyes-Sierra M and Coello Coello CA (2006) Multiple objective particle swarm optimizers : A survey of the state-of-art. *International Journal of Computational Intelligence Research* **2**(3), 287–308.

Schonlau M (1997) *Computer Experiments and Global Optimization*, PhD thesis, University of Waterloo.

Schonlau M, Welch W and Jones D (1998) Global versus local search in constrained optimization of computer models. In *New Developments and Applications in Experimental Design* (eds Flournoy N, Rosenberger WF and Wong WK). Institute of Mathematical Statistics, Hayward, CA, pp. 11–25.

Sierra MR and Coello Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. In *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization* (eds Coello Coello CA, Hernández Aguirre A and Zitzler E), vol. 3410 of *EMO'05*. Springer-Verlag, Berlin, pp. 505–519.

Toscano-Pulido G and Coello Coello CA (2004) Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. In *Proceedings of Genetic and Evolutionary Computation*, vol. 3102 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 225–237.

Toscano-Pulido G, Coello Coello CA and Santana-Quintero LV (2007) EMOPSO: A multi-objective particle swarm optimizer with emphasis on efficiency. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization* (ed. Obayashi S, Deb K, Poloni C, Hiroyasu T and Murata T), vol. 4403 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, pp. 272–285.

Tulkens H (1993) On FDH efficiency: Some methodological issues and applications to retail banking, courts, and urban transit. *Journal of Productivity Analysis* **4**, 183–210.

Yun YB, Nakayama H and Arakawa M (2004a) Multiple criteria decision making with generalized DEA and an aspiration level method. *European Journal of Operational Research* **158**(3), 697–706.

Yun YB, Nakayama H and Tanino T (2000) On efficiency of data envelopment analysis. In *Research and Practice in Multiple Criteria Decision Making* (eds Haimes YY and Steuer RE). Springer-Verlag, Berlin, pp. 208–217.

Yun YB, Nakayama H and Tanino T (2004b) A generalized model for data envelopment analysis. *European Journal of Operational Research* **157**(1), 87–105.

Yun YB, Nakayama H, Tanino T and Arakawa M (2001) Generation of efficient frontiers in multi-objective optimization problems by generalized data envelopment analysis. *European Journal of Operational Research* **129**(3), 586–595.

Zitzler E, Laumanns M and Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich.

10

Fuzzy multiobjective optimization

Masatoshi Sakawa
Department of System Cybernetics, Graduate School of Engineering, Hiroshima University, Japan

10.1 Introduction

In actual decision making situations, we must often make a decision on the basis of vague information or uncertain data. For such decision making problems involving uncertainty, there exist two typical approaches: stochastic programming and fuzzy programming.

Stochastic programming, as an optimization method on the basis of probability theory, has been developing in various ways (Stancu-Minasian 1984, 1990), including a two-stage problem by Dantzig (1955), and chance constrained programming by Charnes and Cooper (1959). Especially, for multiobjective stochastic linear programming problems, Stancu-Minasian (1984, 1990) considered the minimum risk approach, while Leclercq (1982) and Teghem *et al*. (1986) proposed interactive methods.

On the other hand, fuzzy mathematical programming representing the vagueness in decision making situations by fuzzy concepts has been studied by many researchers (Lai and Hwang 1994; Rommelfanger 1996; Sakawa 1993, 2001, 2002). Fuzzy multiobjective linear programming, first proposed by Zimmermann (1978), has been also developed by numerous researchers, and an increasing number of successful applications have been appearing (Delgado *et al*. 1994; Kacprzyk and Orlovski 1987; Kahraman 2008; Lai and Hwang 1994; Luhandjula 1987; Sakawa 1993, 2000, 2001, 2002; Sakawa *et al*.

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

1987b; Slowinski 1998; Slowinski and Teghem 1990; Verdegay and Delgado 1989; Zimmermann 1987).

As a hybrid of the stochastic approach and the fuzzy one, Hulsurkar *et al*. (1997) applied fuzzy programming to multiobjective stochastic linear programming problems. Unfortunately, however, in their method, since membership functions for the objective functions are supposed to be aggregated by a minimum operator or a product operator, optimal solutions which sufficiently reflect the decision maker's preference may not be obtained. Realizing such drawbacks, Sakawa and his colleagues incorporated the techniques of an interactive fuzzy satisficing method which was originally developed for deterministic problems (Sakawa 1993; Sakawa and Yano 1985b) into multiobjective stochastic programming problems, through the introduction of several fuzzy multiobjective stochastic programming models based on different optimization criteria such as expectation optimization (Sakawa and Kato 2002, 2008; Sakawa *et al*. 2003), variance minimization (Sakawa and Kato 2008; Sakawa *et al*. 2002), probability maximization (Sakawa and Kato 2002, 2008; Sakawa *et al*. 2004) and fractile criterion optimization (Sakawa and Kato 2008; Sakawa *et al*. 2001a), to derive a satisficing solution for a decision maker (DM) from the extended Pareto optimal solution sets.

Under these circumstances, in this chapter, multiobjective linear programming and interactive multiobjective linear programming, both incorporating fuzzy goals of the DM, are introduced. Multiobjective linear programming problems with fuzzy parameters are also formulated and interactive decision making methods, both without and with the fuzzy goals of the DM, for deriving a satisficing solution for the DM efficiently are presented. Finally, multiobjective linear programming problems involving random variable coefficients, called stochastic multiobjective linear programming problems, are considered. By making use of stochastic models such as a probability maximization model together with chance constrained conditions, the stochastic multiobjective programming problems are transformed into deterministic ones. After determining the fuzzy goals of the DM, interactive fuzzy satisficing methods are introduced for deriving a satisficing solution for the DM.

10.2 Solution concepts for multiobjective programming

The problem to optimize multiple conflicting objective functions simultaneously under given constraints is called the multiobjective programming problem and can be formulated as the following vector-minimization problem:

$$\left.\begin{array}{ll} \text{minimize} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \\ \text{subject to} & \mathbf{x} \in X = \{\mathbf{x} \in R^n \mid g_j(\mathbf{x}) \le \mathbf{0}, \quad j = 1, \ldots, m\} \end{array}\right\} \tag{10.1}$$

where $f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})$ are k distinct objective functions of the decision vector $\mathbf{x}$, $g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$ are m inequality constraints and X is the feasible set of constrained decisions.

If we directly apply the notion of optimality for single-objective linear programming to this multiobjective programming, we arrive at the following notion of a complete optimal solution.

Definition 10.2.1 (Complete optimal solution) $\mathbf{x}^*$ *is said to be a complete optimal solution if and only if there exists* $\mathbf{x}^* \in X$ *such that* $f_i(\mathbf{x}^*) \le f_i(\mathbf{x})$, $i = 1, \ldots, k$, *for all* $\mathbf{x} \in X$.

However, in general, such a complete optimal solution that simultaneously minimizes all of the multiple objective functions does not always exist when the objective functions conflict with each other. Thus, instead of a complete optimal solution, a new solution concept, called Pareto optimality, is introduced in multiobjective programming (Chankong and Haimes 1983; Sakawa 1993; Steuer 1986).

Definition 10.2.2 (Pareto optimal solution) $\mathbf{x}^* \in X$ *is said to be a Pareto optimal solution if and only if there does not exist another* $\mathbf{x} \in X$ *such that* $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ *for all* $i = 1, \ldots, k$, *and* $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ *for at least one* $j \in \{1, \ldots, k\}$.

As can be seen from the definition, a Pareto optimal solution consists of an infinite number of points. A Pareto optimal solution is sometimes called a noninferior solution since it is not inferior to other feasible solutions.

In addition to Pareto optimality, the following weak Pareto optimality is defined as a slightly weaker solution concept than Pareto optimality.

Definition 10.2.3 (Weak Pareto optimal solution) $\mathbf{x}^* \in X$ *is said to be a weak Pareto optimal solution if and only if there does not exist another* $\mathbf{x} \in X$ *such that* $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$, $1, \ldots, k$.

For notational convenience, let X^{CO}, X^{P}, or X^{WP} denote complete optimal, Pareto optimal, or weak Pareto optimal solution sets, respectively. Then from their definitions, it can be easily understood that the following relation holds:

$$X^{CO} \subseteq X^{P} \subseteq X^{WP}.$$

Several computational methods have been proposed for characterizing Pareto optimal solutions depending on the different methods to scalarize the multiobjective programming problems. Among the many possible ways of scalarizing the multiobjective programming problems, the weighting method, the constraint method, and the weighted minimax method have been studied as a means of characterizing Pareto optimal solutions of the multiobjective programming problems.

The details of multiobjective programming can be found in standard texts including Chankong and Haimes (1983), Steuer (1986), and Sakawa (1993).

10.3 Interactive multiobjective linear programming

The STEP method (STEM) proposed by Benayoun *et al*. (1971) seems to be known as one of the first interactive multiobjective linear programming techniques, but there have been some modifications and extensions (see e.g., Choo and Atkins 1980; Fichefet 1976). Essentially, the STEM algorithm consists of two major steps. Step 1 seeks a Pareto optimal solution that is near to the ideal point in the minimax sense. Step 2 requires the DM to compare the objective vector with the ideal vector and to indicate which objectives can be sacrificed, and by how much, in order to improve the current levels of unsatisfactory objectives. The STEM algorithm is quite simple to understand and implement, in the sense that the DM is required to give only the amounts to be sacrificed of some satisfactory objectives until all objectives become satisfactory. However, the DM will never arrive at the final solution if the DM is not willing to sacrifice any

of the objectives. Moreover, in many practical situations, the DM will probably want to indicate directly the aspiration level for each objective rather than just specify the amount by which satisfactory objectives can be sacrificed.

Wierzbicki (1980) developed a relatively practical interactive method called the reference point method (RPM) by introducing the concept of a reference point suggested by the DM which reflects in some sense the desired values of the objective functions. The basic idea behind the RPM is that the DM can specify reference values for the objective functions and change the reference objective levels interactively due to learning or improved understanding during the solution process. In this procedure, when the DM specifies a reference point, the corresponding scalarization problem is solved for generating the Pareto optimal solution which is, in a sense, close to the reference point or better than that if the reference point is attainable. Then the DM either chooses the current Pareto optimal solution or modifies the reference point to find a satisficing solution.

Since then, some similar interactive multiobjective programming methods have been developed along this line (see, e.g., Steuer and Choo 1983). However, it is important to point out here that for dealing with the fuzzy goals of the DM for each of the objective functions of the multiobjective linear programming problem, Sakawa *et al*. (1987b) developed the extended fuzzy version of the RPM that supplies the DM with the trade-off information even if the fuzzy goals of the DM are not considered. Although the method will be outlined in the next section, it would certainly be appropriate to discuss here the RPM with trade-off information rather than the RPM proposed by Wierzbicki.

Consider the following multiobjective linear programming problem:

$$\left.\begin{array}{ll}\text{minimize} & z_1(\mathbf{x}) = \mathbf{c}_1\mathbf{x} \\ \text{minimize} & z_2(\mathbf{x}) = \mathbf{c}_2\mathbf{x} \\ & \qquad\vdots \\ \text{minimize} & z_k(\mathbf{x}) = \mathbf{c}_k\mathbf{x} \\ \text{subject to} & \mathbf{x} \in X = \{\mathbf{x} \in R^n \mid A\mathbf{x} \leq \mathbf{b},\ \ \mathbf{x} \geq 0\},\end{array}\right\} \tag{10.2}$$

where

$$\begin{aligned} \mathbf{c}_i &= (c_{i1}, \ldots, c_{in}),\ \ i = 1, \ldots, k, \\ x &= (x_1, \ldots, x_n)^T, \\ A &= \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix}, \\ \mathbf{b} &= (b_1, \ldots, b_m)^T. \end{aligned} \tag{10.3}$$

For each of the conflicting objective functions $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \ldots, z_k(\mathbf{x}))^T$, assume that the DM can specify the so-called reference point $\bar{\mathbf{z}} = (\bar{z}_1, \ldots, \bar{z}_k)^T$ which reflects in some sense the desired values of the objective functions of the DM. Also assume that the DM can change the reference point interactively due to learning or improved understanding during the solution process. When the DM specifies the reference point $\bar{\mathbf{z}} = (\bar{z}_1, \ldots, \bar{z}_k)^T$, the corresponding Pareto optimal solution, which is, in the minimax

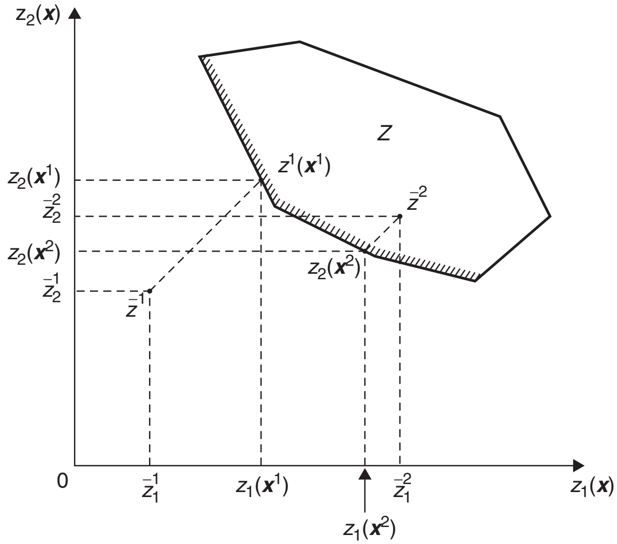

Figure 10.1 Graphical interpretation of minimax method.

sense, nearest to the reference point or better than that if the reference point is attainable, is obtained by solving the following minimax problem:

$$\left.\begin{array}{ll} \text{minimize} & \max_{i=1,\dots,k}\{z_i(\mathbf{x}) - \bar{z}_i\} \\ \text{subject to} & \mathbf{x} \in X, \end{array}\right\} \tag{10.4}$$

or equivalently

$$\left.\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & z_i(\mathbf{x}) - \bar{z}_i \le v, \quad i = 1, \dots, k \\ & \mathbf{x} \in X. \end{array}\right\} \tag{10.5}$$

The case of the two-objective functions in the z_1-z_2 plane is shown geometrically in Figure 10.1. For the two reference points $\bar{\mathbf{z}}^1 = (\bar{z}_1^1, \bar{z}_2^1)^T$ and $\bar{\mathbf{z}}^2 = (\bar{z}_1^2, \bar{z}_2^2)^T$ specified by the DM, solving the corresponding minimax problems yields the corresponding Pareto optimal solutions $\mathbf{z}^1(\mathbf{x}^1)$ and $\mathbf{z}^2(\mathbf{x}^2)$.

The relationships between the optimal solutions of the minimax problem and the Pareto optimal concept of the multiobjective linear programming can be characterized by the following two theorems.

Theorem 10.3.1 (Minimax problem and Pareto optimality) *If* $\mathbf{x}^* \in X$ *is a unique optimal solution of the minimax problem for any reference point* $\bar{\mathbf{z}}$*, then* $\mathbf{x}^*$ *is a Pareto optimal solution of the multiobjective linear programming problem.*

It should be noted that only weak Pareto optimality is guaranteed if the uniqueness of a solution is not guaranteed.

Theorem 10.3.2 (Pareto optimality and minimax problem) *If* $\mathbf{x}^*$ *is a Pareto optimal solution of the multiobjective linear programming problem, then* $\mathbf{x}^*$ *is an optimal solution of the minimax problem for some reference point* $\bar{\mathbf{z}}$.

Now, given the Pareto optimal solution for the reference point specified by the DM by solving the corresponding minimax problem, the DM must either be satisfied with the current Pareto optimal solution or modify the reference point. To help the DM express a degree of preference, trade-off information between a standing objective function $z_1(\mathbf{x})$ and each of the other objective functions is very useful. Such a trade-off between $z_1(\mathbf{x})$ and $z_i(\mathbf{x})$ for each $i = 2, \dots, k$ is easily obtainable since it is closely related to the strict positive simplex multipliers of the minimax problem. Let the simplex multipliers associated with the constraints of the minimax problem be denoted by $\pi_i, \quad i = 1, \dots, k$. If all $\pi_i > 0$ for each i, it can be proved that the following expression holds:

$$-\frac{\partial z_1(\mathbf{x})}{\partial z_i(\mathbf{x})} = \frac{\pi_1}{\pi_i}, \quad i = 2, \dots, k. \tag{10.6}$$

We can now construct the interactive algorithm to derive the satisficing solution for the DM from the Pareto optimal solution set. The steps marked with an asterisk involve interaction with the DM. Observe that this interactive multiobjective linear programming method can be interpreted as the RPM with trade-off information.

Interactive multiobjective linear programming

Step 0: Calculate the individual minimum $z_i^{\min} = \min_{\mathbf{x} \in X} z_i(\mathbf{x})$ and maximum $z_i^{\max} = \max_{\mathbf{x} \in X} z_i(\mathbf{x})$ of each objective function under the given constraints.

Step 1*: Ask the DM to select the initial reference point by considering the individual minimum and maximum. If the DM finds it difficult or impossible to identify such a point, ideal point $z_i^{\min} = \min_{\mathbf{x} \in X} z_i(\mathbf{x})$ can be used for that purpose.

Step 2: For the reference point specified by the DM, solve the corresponding minimax problem to obtain the Pareto optimal solution together with the trade-off rate information between the objective functions.

Step 3*: If the DM is satisfied with the current levels of the Pareto optimal solution, stop. Then the current Pareto optimal solution is the satisficing solution for the DM. Otherwise, ask the DM to update the current reference point by considering the current values of the objective functions together with the trade-off rates between the objective functions and return to Step 2.

It should be stressed to the DM that any improvement of one objective function can be achieved only at the expense of at least one of the other objective functions.

Further details of the theory, methods and applications of interactive multiobjective programming can be found in Chankong and Haimes (1983), Steuer (1986), Sakawa (1993, 2000, 2001, 2002), and Sakawa *et al*. (2011).

The recently published book entitled ‘Multiple Criteria Decision Making – From Early History to the 21st Century’ (Köksalan *et al*. 2011), which begins with the early history of multiple criteria decision making and proceeds to give a decade-by-decade

account of major developments in the field starting from the 1970s until now, would be very useful for interested readers.

10.4 Fuzzy multiobjective linear programming

In 1978, Zimmermann (1978) extended his fuzzy linear programming approach (Zimmermann 1976) to the following multiobjective linear programming problem with k linear objective functions $z_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x}$, $i = 1, \ldots, k$:

$$\left.\begin{array}{ll} \text{minimize} & \mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_k(\mathbf{x}))^T \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \end{array}\right\} \tag{10.7}$$

where $\mathbf{c}_i = (c_{i1}, \ldots, c_{in})$, $i = 1, \ldots, k$, $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathbf{b} = (b_1, \ldots, b_m)^T$ and $A = [a_{ij}]$ is an $m \times n$ matrix.

For each of the objective functions $z_i(\mathbf{x}) = \mathbf{c}_i\mathbf{x}$, $i = 1, \ldots, k$, of this problem, assume that the DM has a fuzzy goal such as 'the objective function $z_i(\mathbf{x})$ should be substantially less than or equal to some value p_i.' Then the corresponding linear membership function $\mu_i^L(z_i(\mathbf{x}))$ is defined as

$$\mu_i^L(z_i(\mathbf{x})) = \begin{cases} 0 & ; z_i(\mathbf{x}) \geq z_i^0 \\ \dfrac{z_i(\mathbf{x}) - z_i^0}{z_i^1 - z_i^0} & ; z_i^0 \geq z_i(\mathbf{x}) \geq z_i^1 \\ 1 & ; z_i(\mathbf{x}) \leq z_i^1 \end{cases} \tag{10.8}$$

where z_i^0 or z_i^1 denotes the value of the objective function $z_i(\mathbf{x})$ such that the degree of membership function is 0 or 1, respectively.

Figure 10.2 illustrates the graph of the possible shape of the linear membership function.

Using such linear membership functions $\mu_i^L(z_i(\mathbf{x}))$, $i = 1, \ldots, k$, and following the fuzzy decision of Bellman and Zadeh (1970), the original multiobjective linear programming problem can be interpreted as

$$\left.\begin{array}{lll} \text{maximize} & \displaystyle\min_{i=1,\ldots,k} & \{\mu_i^L(z_i(\mathbf{x}))\} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b}, & \mathbf{x} \geq \mathbf{0}. \end{array}\right\} \tag{10.9}$$

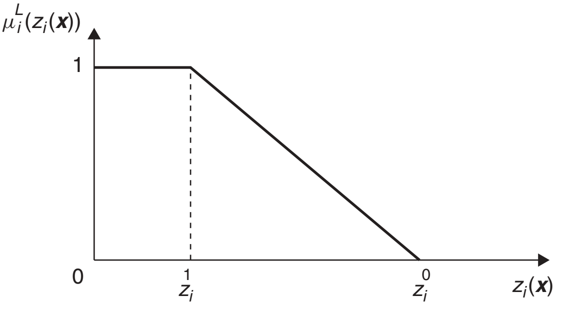

Figure 10.2 Linear membership function.

By introducing the auxiliary variable λ, it can be reduced to the following conventional linear programming problem:

$$\left.\begin{array}{ll} \text{maximize} & \lambda \\ \text{subject to} & \lambda \leq \mu_i^L(z_i(\mathbf{x})), \quad i = 1, 2, \ldots, k \\ & A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq 0. \end{array}\right\} \tag{10.10}$$

By assuming the existence of the optimal solution $\mathbf{x}^{io}$ of the individual objective function minimization problem under the constraints defined by

$$\min_{\mathbf{x} \in X} z_i(\mathbf{x}), \quad i = 1, 2, \ldots, k, \tag{10.11}$$

Zimmermann (1978) suggested a way to determine the linear membership function $\mu_i^L(z_i(\mathbf{x}))$. To be more specific, using the individual minimum

$$z_i^{\min} = z_i(\mathbf{x}^{io}) = \min_{\mathbf{x} \in X} z_i(\mathbf{x}), \quad i = 1, 2, \ldots, k, \tag{10.12}$$

together with

$$z_i^m = \max(z_i(\mathbf{x}^{1o}), \ldots, z_i(\mathbf{x}^{i-1,o}), z_i(\mathbf{x}^{i+1,o}), \ldots, z_i(\mathbf{x}^{ko})), \quad i = 1, 2, \ldots, k, \tag{10.13}$$

he determined the linear membership function as in (10.8) by choosing $z_i^1 = z_i^{\min}$ and $z_i^0 = z_i^m$. For this membership function, it can be easily shown that if the optimal solution of (10.9) or (10.10) is unique, it is also a Pareto optimal solution of the multiobjective linear programming problem.

In the case where not only fuzzy goals but also fuzzy constraints exist, using linear membership functions for fuzzy constraints, there can be a similar discussion.

Zimmermann (1978) called the fuzzy decision the minimum operator, and for other aggregation patterns than the minimum operator, he considered the product fuzzy decision. He called the product fuzzy decision the product operator. Unfortunately, with the product operator, even if we use the linear membership functions, the objective function of this problem becomes a nonlinear function, and hence, the linear programming method (Dantzig 1961) cannot be applied.

In 1983, to quantify the fuzzy goals of the DM by eliciting the corresponding membership functions, Sakawa (1983) proposed using five types of membership functions: linear, exponential, hyperbolic, hyperbolic inverse, and piecewise linear. Through the use of these membership functions including nonlinear ones, the fuzzy goals of the DM are quantified. Then following the fuzzy decision of Bellman and Zadeh (1970), the problem becomes a nonlinear programming problem. However, it can be reduced to a set of linear inequalities if some variable is fixed. Based on this idea, Sakawa (1983) proposed a new method combining the use of the bisection method and the linear programming method (Dantzig 1961).

10.5 Interactive fuzzy multiobjective linear programming

In the fuzzy approaches to multiobjective linear programming problems proposed by Zimmermann (1978) and his successors (Hannan 1981; Leberling 1981; Zimmermann 1983), it has been implicitly assumed that the fuzzy decision of Bellman and Zadeh

(1970) is the proper representation of the fuzzy preferences of the DM. Therefore, these approaches are preferable only when the DM feels that the fuzzy decision is appropriate when combining the fuzzy goals and/or constraints. However, such situations seem to occur rarely in practice and consequently it becomes evident that an interaction with the DM is necessary.

In this section, assuming that the DM has a fuzzy goal for each of the objective functions in multiobjective linear programming problems, we present an interactive fuzzy multiobjective linear programming method incorporating the desirable features of the interactive approaches into the fuzzy approaches.

Fundamental to the multiobjective linear programming is the concept of Pareto optimal solutions, also known as a noninferior solution.

However, considering the imprecise nature inherent in human judgments in multiobjective linear programming problems, the DM may have a fuzzy goal expressed as '$z_i(\mathbf{x})$ should be substantially less than or equal to some value p_i.'

In a minimization problem, a fuzzy goal stated by the DM may be to achieve 'substantially less than or equal to p_i.' This type of statement can be quantified by eliciting a corresponding membership function.

To elicit a membership function $\mu_i(z_i(\mathbf{x}))$ from the DM for each of the objective functions $z_i(\mathbf{x})$, $i = 1, \ldots, k$, we first calculate the individual minimum $z_i^{\min} = \min_{\mathbf{x} \in X} \; z_i(\mathbf{x})$ and maximum $z_i^{\max} = \max_{\mathbf{x} \in X} \; z_i(\mathbf{x})$ of each objective function $z_i(\mathbf{x})$ under the given constraints.

Taking into account the calculated individual minimum and maximum of each objective function together with the rate of increase of membership of satisfaction, the DM must determine the subjective membership function $\mu_i(z_i(\mathbf{x}))$, which is a strictly monotone decreasing function with respect to $z_i(\mathbf{x})$. Here, it is assumed that $\mu_i(z_i(\mathbf{x})) = 0$ or $\to 0$ if $z_i(\mathbf{x}) \geq z_i^0$ and $\mu_i(z_i(\mathbf{x})) = 1$ or $\to 1$ if $z_i(\mathbf{x}) \leq z_i^1$.

So far, we have restricted ourselves to a minimization problem and consequently assumed that the DM has a fuzzy goal such as '$z_i(\mathbf{x})$ should be substantially less than or equal to p_i.' In the fuzzy approaches, however, we can further treat a more general multiobjective linear programming problem in which the DM has two types of fuzzy goals expressed in words such as '$z_i(\mathbf{x})$ should be in the vicinity of r_i' (called fuzzy equal), '$z_i(\mathbf{x})$ should be substantially less than or equal to p_i' (called fuzzy min) or '$z_i(\mathbf{x})$ should be substantially greater than or equal to q_i' (called fuzzy max).

Such a generalized multiobjective linear programming problem may now be expressed as

$$\left.\begin{array}{lll} \text{fuzzy min} & z_i(\mathbf{x}) & i \in I_1 \\ \text{fuzzy max} & z_i(\mathbf{x}) & i \in I_2 \\ \text{fuzzy equal} & z_i(\mathbf{x}) & i \in I_3 \\ \text{subject to} & \mathbf{x} \in X & \end{array}\right\} \tag{10.14}$$

where $I_1 \cup I_2 \cup I_3 = \{1, \ldots, k\}$, $I_i \cap I_j = \emptyset$, $i, j = 1, 2, 3$, $i \neq j$.

Here 'fuzzy min $z_i(\mathbf{x})$' or 'fuzzy max $z_i(\mathbf{x})$' represents the fuzzy goal of the DM such as '$z_i(\mathbf{x})$ should be substantially less than or equal to p_i or greater than or equal to q_i,' and 'fuzzy equal $z_i(\mathbf{x})$' represents the fuzzy goal such as '$z_i(\mathbf{x})$ should be in the vicinity of r_i.'

Concerning the membership function for the fuzzy goal of the DM such as '$z_i(\mathbf{x})$ should be in the vicinity of r_i,' it is obvious that a strictly monotone increasing function

$d_{iL}(z_i)$, $(i \in I_3)$ and a strictly monotone decreasing function $d_{iR}(z_i)$, $(i \in I_3)$, corresponding to the left and right sides of r_i must be determined through interaction with the DM.

Figure 10.3, Figure 10.4 and Figure 10.5 illustrate possible shapes of the fuzzy min, fuzzy max and fuzzy equal membership functions, respectively.

Having elicited the membership functions $\mu_i(z_i(\mathbf{x}))$, $i = 1, \ldots, k$, from the DM for each of the objective functions $z_i(\mathbf{x})$, $i = 1, \ldots, k$, the multiobjective linear programming problem and/or the generalized multiobjective linear programming problem can be converted into the fuzzy multiobjective optimization problem defined by

$$\underset{\mathbf{x}\in X}{\text{maximize}} \ (\mu_1(z_1(\mathbf{x})), \mu_2(z_2(\mathbf{x})), \ldots, \mu_k(z_k(\mathbf{x}))). \tag{10.15}$$

When the fuzzy equal is included in the fuzzy goals of the DM, it is desirable that $z_i(\mathbf{x})$ should be as close to r_i as possible. Consequently, the notion of Pareto optimal

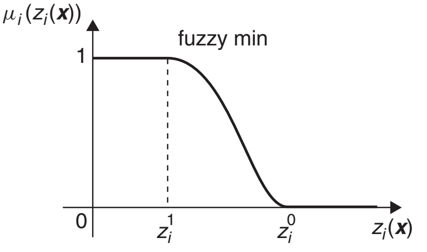

Figure 10.3 Fuzzy min membership function.

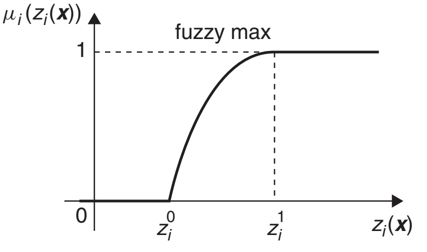

Figure 10.4 Fuzzy max membership function.

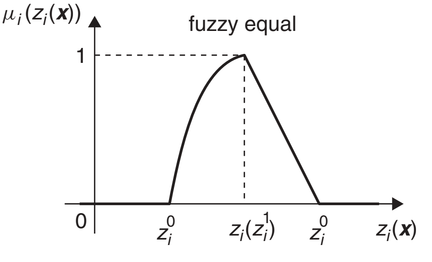

Figure 10.5 Fuzzy equal membership function.

solutions defined in terms of objective functions cannot be applied. For this reason, we introduce the concept of M-Pareto (where M refers to membership) optimal solutions which is defined in terms of membership functions instead of objective functions.

Definition 10.5.1 (M-Pareto optimal solution) $\mathbf{x}^* \in X$ *is said to be an M-Pareto optimal solution to the generalized multiobjective linear programming problem if and only if there does not exist another* $\mathbf{x} \in X$ *such that* $\mu_i(z_i(\mathbf{x})) \geq \mu_i(z_i(\mathbf{x}^*))$ *for all* $i = 1, \ldots, k$, *and* $\mu_j(z_j(\mathbf{x})) > \mu_j(z_j(\mathbf{x}^*))$ *for at least one* $j \in \{1, \ldots, k\}$.

By introducing a general aggregation function

$$\mu_D(\mu(\mathbf{z}(\mathbf{x}))) = \mu_D(\mu_1(z_1(\mathbf{x})), \mu_2(z_2(\mathbf{x})), \ldots, \mu_k(z_k(\mathbf{x}))), \tag{10.16}$$

a general fuzzy multiobjective decision making problem can be defined by

$$\text{maximize}_{\mathbf{x} \in X} \ \mu_D(\mu(\mathbf{z}(\mathbf{x}))). \tag{10.17}$$

Observe that the value of $\mu_D(\mu(\mathbf{z}(\mathbf{x})))$ can be interpreted as representing an overall degree of satisfaction with the DM's multiple fuzzy goals.

Probably the most crucial problem in the fuzzy multiobjective decision making problem is the identification of an appropriate aggregation function which well represents the DM's fuzzy preferences. If $\mu_D(\cdot)$ can be explicitly identified, then the fuzzy multiobjective decision making problem reduces to a standard mathematical programming problem. However, this rarely happens, and as an alternative, an interaction with the DM is necessary for finding the satisficing solution of the fuzzy multiobjective decision making problem.

In the interactive fuzzy multiobjective linear programming method proposed by Sakawa *et al*. (1987b), after determining the membership functions $\mu(\mathbf{z}(\mathbf{x})) = (\mu_1(z_1(\mathbf{x})), \ldots, \mu_k(z_k(\mathbf{x})))^T$ for each of the objective functions $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), \ldots, z_k(\mathbf{x}))^T$, for generating a candidate for the satisficing solution which is also M-Pareto optimal, the DM is then asked to specify the aspiration levels of achievement for the membership values of all membership functions, called the reference membership levels. The reference membership levels can be viewed as natural extensions of the reference point of Wierzbicki (1980) in objective function spaces.

For the DM's reference membership levels $\overline{\mu} = (\overline{\mu}_1, \ldots, \overline{\mu}_k)^T$, the corresponding M-Pareto optimal solution, which is nearest to the requirements in the minimax sense or better than that if the reference membership levels are attainable, is obtained by solving the following minimax problem

$$\text{minimize}_{\mathbf{x} \in X} \max_{i=1,\ldots,k} \{\overline{\mu}_i - \mu_i(z_i(\mathbf{x}))\}, \tag{10.18}$$

or equivalently

$$\left.\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & \overline{\mu}_i - \mu_i(z_i(\mathbf{x})) \leq v, \ \ i = 1, \ldots, k \\ & \mathbf{x} \in X. \end{array}\right\} \tag{10.19}$$

The relationships between the optimal solutions of the minimax problem and the M-Pareto optimal concept of the multiobjective linear programming problem can be characterized by the following theorems.

Theorem 10.5.2 (Minimax problem and M-Pareto optimality) *If* $\mathbf{x}^* \in X$ *is a unique optimal solution to the minimax problem for some* $\overline{\mu}_i$, $i = 1, \ldots, k$, *then* $\mathbf{x}^*$ *is an M-Pareto optimal solution to the generalized multiobjective linear programming problem.*

Definition 10.5.3 (M-Pareto optimality and minimax problem) *If* $\mathbf{x}^*$ *is an M-Pareto optimal solution to the generalized multiobjective linear programming problem with* $0 < \mu_i(z_i(\mathbf{x}^*)) < 1$ *holding for all* i, *then there exists* $\overline{\mu}_i$, $i = 1, \ldots, k$, *such that* $\mathbf{x}^*$ *is an optimal solution to the minimax problem.*

If all of the membership functions $\mu_i(z_i(\mathbf{x}))$, $i = 1, \ldots, k$, are linear, the minimax problem becomes a linear programming problem, and hence, we can obtain an optimal solution by directly applying the simplex method of linear programming Dantzig (1961).

However, with the strictly monotone decreasing or increasing membership functions, which may be nonlinear, the resulting minimax problem becomes a nonlinear programming problem. For notational convenience, denote the strictly monotone decreasing function for the fuzzy min and the right function of the fuzzy equal by $d_{iR}(z_i)(i \in I_1 \cup I_3)$ and the strictly monotone increasing function for the fuzzy max and the left function of the fuzzy equal by $d_{iL}(z_i)(i \in I_2 \cup I_3)$. Then in order to solve the formulated problem on the basis of the linear programming method, convert each constraint $\overline{\mu}_i - \mu_i(z_i(\mathbf{x})) \leq v$, $i = 1, \ldots, k$, of the minimax problem (10.19) into the following form using the strictly monotone property of $d_{iL}(\cdot)$ and $d_{iR}(\cdot)$:

$$\left.\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & z_i(\mathbf{x}) \leq d_{iR}^{-1}(\overline{\mu}_i - v), \quad i \in I_1 \cup I_3 \\ & z_i(\mathbf{x}) \geq d_{iL}^{-1}(\overline{\mu}_i - v), \quad i \in I_2 \cup I_3 \\ & \mathbf{x} \in X. \end{array}\right\} \tag{10.20}$$

It is important to note here that, if the value of v is fixed, it can be reduced to a set of linear inequalities. Obtaining the optimal solution v^* to the above problem is equivalent to determining the minimum value of v so that there exists an admissible set satisfying the constraints of (10.20). Since v satisfies $\overline{\mu}_{\max} - 1 \leq v \leq \overline{\mu}_{\max}$, where $\overline{\mu}_{\max}$ denotes the maximum value of $\overline{\mu}_i$, $i = 1, \ldots, k$, we have the following method for solving this problem by combined use of the bisection method and the simplex method of linear programming (Dantzig 1961). Here, when $\overline{\mu}_i - v \leq 0$, set $\overline{\mu}_i - v = 0$ in view of the constraints $\overline{\mu}_i - v \leq \mu_i(z_i(\mathbf{x}))$ for $0 \leq \mu_i(z_i(\mathbf{x})) \leq 1$, $i = 1, \ldots, k$.

Step 1: Set $v = \overline{\mu}_{\max}$ and test whether an admissible set satisfying the constraints of (10.20) exists or not using phase one of the simplex method. If an admissible set exists, proceed. Otherwise, the DM must reassess the membership function.

Step 2: Set $v = \overline{\mu}_{\max} - 1$ and test whether an admissible set satisfying the constraints of (10.20) exists or not using phase one of the simplex method. If an admissible set exists, set $v^* = \overline{\mu}_{\max} - 1$. Otherwise, go to the next step since the minimum v which satisfies the constraints of (10.20) exists between $\overline{\mu}_{\max} - 1$ and $\overline{\mu}_{\max}$.

Step 3: For the initial value of $v = \overline{\mu}_{\max} - 0.5$, update the value of v using the bisection method as follows:

$$\begin{cases} v_{n+1} = v_n - 1/2^{n+1} & \text{if an admissible set exists for } v_n, \\ v_{n+1} = v_n + 1/2^{n+1} & \text{if no admissible set exists for } v_n. \end{cases}$$

For each v_n, $n = 1, 2, \ldots$, test whether an admissible set of (10.20) exists or not using the sensitivity analysis technique for changes in the right-hand side of the simplex method and determine the minimum value of v satisfying the constraints of (10.20).

In this way, we can determine the optimal solution v^*. Then the DM selects an appropriate standing objective from among the objectives $z_i(\mathbf{x})$, $i = 1, \ldots, k$. For notational convenience in the following without loss of generality, let it be $z_1(\mathbf{x})$ and $1 \in I_1$. Then the following linear programming problem is solved for $v = v^*$:

$$\left.\begin{array}{ll} \text{minimize} & z_1(\mathbf{x}) \\ \text{subject to} & z_i(\mathbf{x}) \leq d_{iR}^{-1}(\overline{\mu}_i - v^*), \;\; i(\neq 1) \in I_1 \cup I_3 \\ & z_i(\mathbf{x}) \geq d_{iL}^{-1}(\overline{\mu}_i - v^*), \;\; i(\neq 1) \in I_2 \cup I_3 \\ & \mathbf{x} \in X. \end{array}\right\} \tag{10.21}$$

The DM must either be satisfied with the current M-Pareto optimal solution or act on this solution by updating the reference membership levels. In order to help the DM express a degree of preference, trade-off information between a standing membership function $\mu_1(z_1(\mathbf{x}))$ and each of the other membership functions is very useful. Such trade-off information is easily obtainable since it is closely related to the simplex multipliers of the problem (10.21).

Let the simplex multipliers corresponding to the constraints $z_i(\mathbf{x})$, $i = 2, \ldots, k$, of the linear problem (10.21) be denoted by $\pi_i^* = \pi_i(\mathbf{x}^*)$, $i = 2, \ldots, k$, where $\mathbf{x}^*$ is an optimal solution of (10.21). If $\mathbf{x}^*$ is a nondegenerate solution of (10.21) and all the constraints of (10.21) are active, then by using the results in Haimes and Chankong (1979), the trade-off information between the objective functions can be represented by

$$-\frac{\partial z_1(\mathbf{x})}{\partial z_i(\mathbf{x})} = \pi_i^*, \;\; i = 2, \ldots, k. \tag{10.22}$$

Hence, by the chain rule, the trade-off information between the membership functions is given by

$$-\frac{\partial \mu_1(z_1(\mathbf{x}))}{\partial \mu_i(z_i(\mathbf{x}))} = -\frac{\partial \mu_1(z_1(\mathbf{x}))}{\partial z_1(\mathbf{x})} \frac{\partial z_1(\mathbf{x})}{\partial z_i(\mathbf{x})} \left\{ \frac{\partial \mu_i(z_i(\mathbf{x}))}{\partial z_i(\mathbf{x})} \right\}^{-1}, \;\; i = 2, \ldots, k. \tag{10.23}$$

Therefore, for each $i = 2, \ldots, k$, we have the following expression:

$$-\frac{\partial \mu_1(z_1(\mathbf{x}))}{\partial \mu_i(z_i(\mathbf{x}))} = \pi_i^* \frac{\partial \mu_1(z_1(\mathbf{x}))/\partial z_1(\mathbf{x})}{\partial \mu_i(z_i(\mathbf{x}))/\partial z_i(\mathbf{x})}, \;\; i = 2, \ldots, k. \tag{10.24}$$

It should be stressed here that in order to obtain the trade-off rate information from (10.24), all the constraints of the problem (10.21), must be active. Therefore, if there are

inactive constraints, it is necessary to replace $\overline{\mu}_i$ for inactive constraints by $\overline{\mu}_i(z_i(\mathbf{x}^*))$ and solve the corresponding problem to obtain the simplex multipliers.

We can now construct the interactive algorithm in order to derive the satisficing solution for the DM from the M-Pareto optimal solution set where the steps marked with an asterisk involve interaction with the DM. This interactive fuzzy multiobjective programming method can also be interpreted as the fuzzy version of the RPM with trade-off information.

Interactive fuzzy multiobjective linear programming

Step 0: Calculate the individual minimum and maximum of each objective function under the given constraints.

Step 1*: Elicit a membership function from the DM for each of the objective functions.

Step 2: Set the initial reference membership levels to 1.

Step 3: For the reference membership levels, solve the corresponding minimax problem to obtain the M-Pareto optimal solution and the membership function value together with the trade-off rate information between the membership functions.

Step 4*: If the DM is satisfied with the current levels of the M-Pareto optimal solution, stop. Then the current M-Pareto optimal solution is the satisficing solution for the DM. Otherwise, ask the DM to update the current reference membership levels by considering the current values of the membership functions together with the trade-off rates between the membership functions and return to Step 3.

It should be stressed to the DM that any improvement of one membership function can be achieved only at the expense of at least one of the other membership functions.

In the next section, we will proceed to the multiobjective linear programming problems with fuzzy parameters as a generalized version of this section.

10.6 Interactive fuzzy multiobjective linear programming with fuzzy parameters

First, recall the multiobjective linear programming problems discussed thus far. For convenience in our subsequent discussion, consider the multiobjective linear programming of the following form:

$$\left.\begin{array}{ll}\text{minimize} & (\mathbf{c}_1\mathbf{x}, \mathbf{c}_2\mathbf{x}, \ldots, \mathbf{c}_k\mathbf{x})^T \\ \text{subject to} & \mathbf{x} \in X = \{\mathbf{x} \in R^n \mid \mathbf{a}_j\mathbf{x} \leq b_j, \quad j = 1, \ldots, m; \quad \mathbf{x} \geq \mathbf{0}\}\end{array}\right\} \tag{10.25}$$

where $\mathbf{x}$ is an n-dimensional column vector of decision variable, $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ are n-dimensional cost factor row vectors, $a_1, a_2, \ldots, a_m$ are n-dimensional constraint row vectors, and $b_1, b_2, \ldots, b_m$ are constants.

In practice, however, it would certainly be more appropriate to consider that the possible values of the parameters in the description of the objective functions and the constraints usually involve the ambiguity of the experts' understanding of the real system. For this reason, in this chapter, we consider the following multiobjective linear programming problem involving fuzzy parameters (MOLP-FP):

$$\left.\begin{array}{ll}\text{minimize} & (\mathbf{c}_1\mathbf{x}, \mathbf{c}_2\mathbf{x}, \ldots, \mathbf{c}_k\mathbf{x})^T \\ \text{subject to} & \mathbf{x} \in X(\mathbf{A}, \mathbf{B}) = \{\mathbf{x} \in R^n \mid \mathbf{A}_j\mathbf{x} \leq B_j, \quad j = 1, \ldots, m; \quad \mathbf{x} \geq \mathbf{0}\}\end{array}\right\} \tag{10.26}$$

$\mathbf{c}_i = (C_{i1}, \ldots, C_{in})$, $\mathbf{A}_j = (A_{j1}, \ldots, A_{jn})$, B_j represent, respectively, fuzzy parameters involved in the objective function $\mathbf{c}_i\mathbf{x}$ and constraint $\mathbf{A}_j\mathbf{x} \leq B_j$.

These fuzzy parameters, reflecting the experts' ambiguous understanding of the nature of the parameters in the problem-formulation process, are assumed to be characterized as fuzzy numbers introduced by Dubois and Prade (1978, 1980).

We now assume that all of the fuzzy parameters $C_{i1}, \ldots, C_{in}$, $A_{j1}, \ldots, A_{jn}$, and B_j in the MOLP-FP are fuzzy numbers whose membership functions are denoted by $\mu_{C_{i1}}(c_{i1}), \ldots, \mu_{C_{in}}(c_{in})$, $\mu_{A_{j1}}(a_{j1}), \ldots, \mu_{A_{jn}}(a_{jn})$, and $\mu_{B_j}(b_j)$, respectively. For simplicity in notation, define the following vectors:

$$\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_k), \quad \mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_m), \quad \mathbf{b} = (b_1, \ldots, b_m),$$

$$\mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_k), \quad \mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_m), \quad \mathbf{B} = (B_1, \ldots, B_m)$$

Observing that the MOLP-FP involves fuzzy numbers both in the objective functions and the constraints, it is evident that the notion of Pareto optimality defined for the multiobjective linear programming cannot be applied directly. Thus, it seems essential to extend the notion of usual Pareto optimality in some sense. For that purpose, we first introduce the α-level set of the fuzzy numbers A_{jr}, B_j, and C_{ir}. To be more explicit, the α-level set of the fuzzy numbers A_{jr}, B_j, and C_{ir} is defined as the ordinary set $(\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ for which the degree of their membership functions exceeds the level α:

$$\begin{aligned}(\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha \; = \; \{ (\mathbf{a}, \mathbf{b}, \mathbf{c}) \mid \mu_{A_{jr}}(a_{jr}) \geq \alpha, \quad \mu_{B_j}(b_j) \geq \alpha, \quad \mu_{C_{ir}}(c_{ir}) \geq \alpha; \\ i = 1, \ldots, k, \quad j = 1, \ldots, m, \quad r = 1, \ldots, n\}\end{aligned} \tag{10.27}$$

Now suppose that the DM decides that the degree of all of the membership functions of the fuzzy numbers involved in the MOLP-FP should be greater than or equal to some value α. Then for such a degree α, the MOLP-FP can be interpreted as the following nonfuzzy multiobjective linear programming [MOLP-FP($\mathbf{a}, \mathbf{b}, \mathbf{c}$)] problem which depends on the coefficient vector $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$:

$$\left.\begin{array}{ll}\text{minimize} & (\mathbf{c}_1\mathbf{x}, \mathbf{c}_2\mathbf{x}, \ldots, \mathbf{c}_k\mathbf{x})^T \\ \text{subject to} & \mathbf{x} \in X(\mathbf{a}, \mathbf{b}) = \{\mathbf{x} \mid R^n \in \mathbf{a}_j\mathbf{x} \leq b_j, \quad j = 1, \ldots, m \quad ; \quad \mathbf{x} \geq \mathbf{0}\}\end{array}\right\} \tag{10.28}$$

Observe that there exists an infinite number of such MOLP-FP($\mathbf{a}, \mathbf{b}, \mathbf{c}$) depending on the coefficient vector $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$, and the values of $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ are arbitrary for any $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ in the sense that the degree of all of the membership functions for the fuzzy numbers in the MOLP-FP exceeds the level α. However, if possible, it

would be desirable for the DM to choose $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ in the MOLP-FP$(\mathbf{a}, \mathbf{b}, \mathbf{c})$ to minimize the objective functions under the constraints. From such a point of view, for a certain degree α, it seems to be quite natural to have the MOLP-FP as the following nonfuzzy α-multiobjective linear programming (α-MOLP) problem:

$$\left.\begin{array}{ll} \text{minimize} & (\mathbf{c}_1\mathbf{x}, \mathbf{c}_2\mathbf{x}, \ldots, \mathbf{c}_k\mathbf{x})^T \\ \text{subject to} & \mathbf{x} \in X(\mathbf{a}, \mathbf{b}) = \{\mathbf{x} \in R^n \mid \mathbf{a}_j\mathbf{x} \leq b_j, \quad j = 1, \ldots, m \quad ; \quad \mathbf{x} \geq \mathbf{0}\} \\ & (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha. \end{array}\right\} \tag{10.29}$$

It should be emphasized here that, in the α-MOLP, the parameters $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ are treated as decision variables rather than constants.

On the basis of the α-level sets of the fuzzy numbers, we can introduce the concept of an α-Pareto optimal solution to the α-MOLP as a natural extension of the Pareto optimality concept for the MOLP.

Definition 10.6.1 (α-Pareto optimal solution) $\mathbf{x}^* \in X(\mathbf{a}^*, \mathbf{b}^*)$ *is said to be an α-Pareto optimal solution to the α-MOLP if and only if there does not exist another* $\mathbf{x} \in X(\mathbf{a}, \mathbf{b})$, $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ *such that* $\mathbf{c}_i\mathbf{x} \leq \mathbf{c}_i^*\mathbf{x}^*$, $i = 1, \ldots, k$ *with strict inequality holding for at least one i, where the corresponding values of parameters* $(\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*)$ *are called α-level optimal parameters.*

Observe that α-Pareto optimal solutions and α-level optimal parameters can be obtained through a direct application of the usual scalarizing methods for generating Pareto optimal solutions by regarding the decision variables in the α-MOLP as $(\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c})$.

As can be seen from the definition of α-Pareto optimality, in general, α-Pareto optimal solutions to the α-MOLP consist of an infinite number of points.

In order to derive a satisficing solution for the DM efficiently from an α-Pareto optimal solution set, interactive programming methods have been presented by Sakawa and Yano (1986, 1990b) and Sakawa (1993).

However, considering the imprecise nature of the DM's judgment, it is natural to assume that the DM may have imprecise or fuzzy goals for each of the objective functions in the α-MOLP. In a minimization problem, a goal stated by the DM may be to achieve 'substantially less than or equal to some value p_i.' This type of statement can be quantified by eliciting a corresponding membership function.

To elicit a membership function $\mu_i(\mathbf{c}_i\mathbf{x})$ from the DM for each of the objective functions $\mathbf{c}_i\mathbf{x}$, $i = 1, \ldots, k$, in the α-MOLP, we first calculate the individual minimum and maximum of each objective function under the given constraints for $\alpha = 0$ and $\alpha = 1$. By taking account of the calculated individual minimum and maximum of each objective function for $\alpha = 0$ and $\alpha = 1$ together with the rate of increase of membership satisfaction, the DM may be able to determine a membership function $\mu_i(\mathbf{c}_i\mathbf{x})$ in a subjective manner which is a strictly monotone decreasing function with respect to $\mathbf{c}_i\mathbf{x}$. So far we have restricted ourselves to a minimization problem and consequently assumed that the DM has a fuzzy goal such as '$\mathbf{c}_i\mathbf{x}$ should be substantially less than or equal to p_i.' In the fuzzy approaches, as discussed previously, we can further treat a more general case where the DM has two types of fuzzy goals, namely, fuzzy goals expressed in words such as '$\mathbf{c}_i\mathbf{x}$ should be in the vicinity of r_i' (called fuzzy equal) as well as '$\mathbf{c}_i\mathbf{x}$ should be substantially less than or equal to p_i or greater than or equal to q_i' (called fuzzy min or

fuzzy max). Such a generalized α-MOLP (Gα-MOLP) problem may now be expressed as

$$\left.\begin{array}{lll} \text{fuzzy min} & \mathbf{c}_i\mathbf{x} & i \in I_1 \\ \text{fuzzy max} & \mathbf{c}_i\mathbf{x} & i \in I_2 \\ \text{fuzzy equal} & \mathbf{c}_i\mathbf{x} & i \in I_3 \\ \text{subject to} & \mathbf{x} \in X(\mathbf{a}, \mathbf{b}) & \\ & (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha & \end{array}\right\} \tag{10.30}$$

where $I_1 \cup I_2 \cup I_3 = \{1, 2, \ldots, k\}$, $I_i \cap I_j = \emptyset$, $i, j = 1, 2, 3$, $i \neq j$.

To elicit a membership function $\mu_i(\mathbf{c}_i\mathbf{x})$ from the DM for a fuzzy goal like '$\mathbf{c}_i\mathbf{x}$ should be in the vicinity of r_i,' it should be quite apparent that different functions can be utilized for both the left and right sides of r_i.Concerning the membership functions of the Gα-MOLP, it is reasonable to assume that $\mu_i(\mathbf{c}_i\mathbf{x})$, $i \in I_1$, and the right side functions of $\mu_i(\mathbf{c}_i\mathbf{x})$, $i \in I_3$, are strictly monotone increasing and continuous functions with respect to $\mathbf{c}_i\mathbf{x}$.

Here it is assumed that $d_{iR}(c_ix)$ is a strictly monotone decreasing continuous function with respect to c_ix and $d_{iL}(c_ix)$ is a strictly monotone increasing continuous function with respect to c_ix. Both may be linear or nonlinear. $(c_ix)_L^0$ and $(c_ix)_R^0$ are maximum values of unacceptable levels for c_ix, and $(c_ix)_L^1$ and $(c_ix)_R^1$ are minimum values of totally desirable levels for c_ix.

When a fuzzy equal is included in the fuzzy goals of the DM, it is desirable that c_ix should be as close to r_i as possible. Consequently, the notion of α-Pareto optimal solutions defined in terms of objective functions cannot be applied. For this reason, we introduce the concept of M-α-Pareto optimal solutions which is defined in terms of membership functions instead of objective functions.

Definition 10.6.2 (M-α-Pareto optimal solution) *A solution $\mathbf{x}^* \in X(\mathbf{a}^*, \mathbf{b}^*)$ is said to be an M-α-Pareto optimal solution to the Gα-MOLP if and only if there does not exist another $\mathbf{x} \in X(\mathbf{a}, \mathbf{b})$, $(\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ such that $\mu_i(\mathbf{c}_i\mathbf{x}) \geq \mu_i(\mathbf{c}_i^*\mathbf{x}^*)$, $i = 1, \ldots, k$, with strict inequality holding for at least one i, where the corresponding values of parameters $(\mathbf{a}^*, \mathbf{b}^*, \mathbf{c}^*) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha$ are called α-level optimal parameters.*

Observe that the concept of M-α-Pareto optimal solutions defined in terms of membership functions is a natural extension to that of α-Pareto optimal solutions defined in terms of objective functions when fuzzy equal is included in the fuzzy goals of the DM.

Having elicited the membership functions $\mu_i(\mathbf{c}_i\mathbf{x})$, $i = 1, \ldots, k$, from the DM for each of the objective functions $\mathbf{c}_i\mathbf{x}$, $i = 1, \ldots, k$, if we introduce a general aggregation function $\mu_D(\cdot)$, a general fuzzy α-multiobjective decision making problem (Fα-MODMP) can be defined by

$$\left.\begin{array}{ll} \text{maximize} & \mu_D(\mu_1(\mathbf{c}_1\mathbf{x}), \mu_2(\mathbf{c}_2\mathbf{x}), \ldots, \mu_k(\mathbf{c}_k\mathbf{x}), \alpha) \\ \text{subject to} & (\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c}) \in P(\alpha), \quad \alpha \in [0, 1] \end{array}\right\} \tag{10.31}$$

where $P(\alpha)$ is the set of M-α-Pareto optimal solutions and corresponding α-level optimal parameters to the Gα-MOLP.

Probably the most crucial problem in the Fα-MODMP is the identification of an appropriate aggregation function which well represents the human DM's fuzzy preferences. If $\mu_D(\cdot)$ can be explicitly identified, then the Fα-MODMP reduces to a standard

mathematical programming problem. However, this happens rarely, and as an alternative approach, it becomes evident that an interaction with the DM is necessary.

To generate a candidate for the satisficing solution, which is also M-α-Pareto optimal, in our decision making method, the DM is asked to specify the degree α of the α-level set and the reference membership values. Observe that the idea of the reference membership values, which first appeared in Sakawa *et al*. (1984), can be viewed as an obvious extension of the idea of the reference point in Wierzbicki (1980).

Once the DM's degree α and reference membership values $\overline{\mu}_i$, $i = 1, \ldots, k$, are specified, the corresponding M-α-Pareto optimal solution, which is, in the minimax sense, nearest to the requirement or better than that if the reference levels are attainable, is obtained by solving the following minimax problem:

$$\min_{\substack{\mathbf{x}\in X(\mathbf{a},\mathbf{b})\\(\mathbf{a},\mathbf{b},\mathbf{c})\in(\mathbf{A},\mathbf{B},\mathbf{C})_\alpha}} \max_{i=1,\ldots,k} (\overline{\mu}_i - \mu_i(\mathbf{c}_i\mathbf{x})) \tag{10.32}$$

or equivalently

$$\left.\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & \overline{\mu}_i - \mu_i(\mathbf{c}_i\mathbf{x}) \le v, \quad i = 1, \ldots, k, \\ & \mathbf{a}_j\mathbf{x} \le b_j, \quad j = 1, \ldots, m, \quad \mathbf{x} \ge \mathbf{0}, \\ & (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in (\mathbf{A}, \mathbf{B}, \mathbf{c})_\alpha. \end{array}\right\} \tag{10.33}$$

However, with the strictly monotone decreasing or increasing membership function, which may be nonlinear, the resulting problem becomes a nonlinear programming problem.

For notational convenience, denote the strictly monotone decreasing function for the fuzzy min and the right function of the fuzzy equal by $d_{iR}(z_i)$ $(i \in I_1 \cup I_3)$ and the strictly monotone increasing function for the fuzzy max and the left function of the fuzzy equal by $d_{iL}(z_i)$ $(i \in I_2 \cup I_3)$. Then in order to solve the formulated problem on the basis of the linear programming method, we first convert each constraint $\overline{\mu}_i - \mu_i(\mathbf{c}_i\mathbf{x}) \le v$, $i = 1, \ldots, k$, of the minimax problem (10.33) into the following form using the strictly monotone property of $d_{iL}(\cdot)$ and $d_{iR}(\cdot)$:

$$\left.\begin{array}{ll} \mathbf{c}_i\mathbf{x} \le d_{iR}^{-1}(\overline{\mu}_i - v), & i \in I_1 \cup I_{3R} \\ \mathbf{c}_i\mathbf{x} \ge d_{iL}^{-1}(\overline{\mu}_i - v), & i \in I_2 \cup I_{3L} \end{array}\right\} \tag{10.34}$$

Now we can introduce the following set-valued functions $S_{iR}(\cdot)$, $S_{iL}(\cdot)$, and $T_j(\cdot,\cdot)$:

$$\left.\begin{array}{ll} S_{iR}(\mathbf{c}_i) = \{(\mathbf{x}, v) \mid \mathbf{c}_i\mathbf{x} \le d_{iR}^{-1}(\overline{\mu}_i - v)\}, & i \mid I_1 \cup I_{3R} \\ S_{iL}(\mathbf{c}_i) = \{(\mathbf{x}, v) \mid \mathbf{c}_i\mathbf{x} \ge d_{iL}^{-1}(\overline{\mu}_i - v)\}, & i \in I_2 \cup I_{3L} \\ T_j(\mathbf{a}_j, b_j) = \{\mathbf{x} \in \mathbf{a}_j\mathbf{x} \le b_j\}, \quad j = 1, \ldots, m & \end{array}\right\} \tag{10.35}$$

It can be verified that the following relations hold for $S_{iR}(\cdot)$, $S_{iL}(\cdot)$, and $T_j(\cdot,\cdot)$ when $\mathbf{x} \ge \mathbf{0}$.

Proposition 10.6.3 (Inclusion relations of set-valued functions)

(1) If $c_i^1 \leq c_i^2$, *then* $S_{iR}(c_i^1) \supseteq S_{iR}(c_i^2)$ *and* $S_{iL}(c_i^1) \subseteq S_{iL}(c_i^2)$.

(2) If $a_j^1 \leq a_j^2$, *then* $T_j(a_j^1, b_j) \supseteq T_j(a_j^2, b_j)$.

(3) If $b_j^1 \leq b_j^2$, *then* $T_j(a_j, b_j^1) \subseteq T_j(a_j, b_j^2)$.

Using the properties of the α-level sets for the vectors of the fuzzy numbers C_i, A_j and the fuzzy numbers B_j, the feasible regions for c_i, a_j, and b_j can be denoted, respectively, by the closed intervals $[\mathbf{c}_{i\alpha}^L, \mathbf{c}_{i\alpha}^R]$, $[\mathbf{a}_{j\alpha}^L, \mathbf{a}_{j\alpha}^R]$, and $[b_{j\alpha}^L, b_{j\alpha}^R]$.

Consequently, using the results in Proposition 10.6, we can obtain an optimal solution to (10.33) by solving the following problem:

$$\left.\begin{array}{ll} \text{minimize} & v \\ \text{subject to} & \mathbf{c}_{i\alpha}^L \mathbf{x} \leq d_{iR}^{-1}(\overline{\mu}_i - v), \quad i \in I_1 \cup I_{3R} \\ & \mathbf{c}_{i\alpha}^R \mathbf{x} \geq d_{iL}^{-1}(\overline{\mu}_i - v), \quad i \in I_2 \cup I_{3L} \\ & \mathbf{a}_{j\alpha}^L \mathbf{x} \leq b_{j\alpha}^R, \quad j = 1, \ldots, m, \quad \mathbf{x} \geq \mathbf{0} \end{array}\right\} \tag{10.36}$$

It is important to note here that this formulation, if the value of v is fixed, can be reduced to a set of linear inequalities. Obtaining the optimal solution v^* to the above problem is equivalent to determining the minimum value of v so that there exists an admissible set satisfying the constraints of (10.36). Since v satisfies $\overline{\mu}_{\max} - 1 \leq v \leq \overline{\mu}_{\max}$, where $\overline{\mu}_{\max}$ denotes the maximum value of $\overline{\mu}_i$, $i = 1, \ldots, k$, we have the following method for solving this problem by combined use of the bisection method and the simplex method of linear programming (Dantzig 1961).

Step 1: Set $v = \overline{\mu}_{\max}$ and test whether an admissible set satisfying the constraints of (10.36) exists or not by making use of phase one of the simplex method. If an admissible set exists, proceed. Otherwise, the DM must reassess the membership function.

Step 2: Set $v = \overline{\mu}_{\max} - 1$ and test whether an admissible set satisfying the constraints of (10.36) exists or not using phase one of the simplex method. If an admissible set exists, set $v^* = \overline{\mu}_{\max} - 1$. Otherwise, go to the next step since the minimum v which satisfies the constraints of (10.36) exists between $\overline{\mu}_{\max} - 1$ and $\overline{\mu}_{\max}$.

Step 3: For the initial value of $v = \overline{\mu}_{\max} - 0.5$, update the value of v using the bisection method as follows:

$$\begin{cases} v_{n+1} = v_n - 1/2^{n+1} & \text{if an admissible set exists for } v_n, \\ v_{n+1} = v_n + 1/2^{n+1} & \text{if no admissible set exists for } v_n. \end{cases} \tag{10.37}$$

For each v_n, $n = 1, 2, \ldots$, test whether an admissible set of (10.36) exists or not using the sensitivity analysis technique for the changes in the right-hand side of the simplex method and determine the minimum value of v satisfying the constraints of (10.36).

In this way, we can determine the optimal solution v^*. Then the DM selects an appropriate standing objective from among the objectives $c_i x$, $i = 1, \ldots, k$. For notational convenience in the following without loss of generality, let it be $c_1 x$ and $1 \in I_1$. Then the following linear programming problem is solved for $v = v^*$:

$$\left.\begin{array}{ll} \text{minimize} & \mathbf{c}_{1\alpha}^{L}\mathbf{x} \\ \text{subject to} & \mathbf{c}_{i\alpha}^{L}\mathbf{x} \le d_{iR}^{-1}(\overline{\mu}_i - v^*), \quad i \in I_1 \cup I_{3R}, \\ & \mathbf{c}_{i\alpha}^{R}\mathbf{x} \ge d_{iL}^{-1}(\overline{\mu}_i - v^*), \quad i \in I_2 \cup I_{3L}, \\ & \mathbf{a}_{j\alpha}^{L}\mathbf{x} \le b_{j\alpha}^{R}, \quad j = 1, \ldots, m, \quad \mathbf{x} \ge \mathbf{0} \end{array}\right\} \tag{10.38}$$

For convenience in our subsequent discussion, we assume that the optimal solution x^* to (10.38) satisfies the following conditions:

$$\left.\begin{array}{ll} \mathbf{c}_{i\alpha}^{L}\mathbf{x}^* = d_{iR}^{-1}(\overline{\mu}_i - v^*), & i \in I_1 \cup I_{3R}, \\ \mathbf{c}_{i\alpha}^{R}\mathbf{x}^* = d_{iL}^{-1}(\overline{\mu}_i - v^*), & i \in I_2 \cup I_{3L} \end{array}\right\} \tag{10.39}$$

where $I_3 = I_{3L} \cup I_{3R}$ and $I_{3L} \cap I_{3R} = \emptyset$.

It is interesting to note that $c_{i\alpha}^{L}, i \in I_1 \cup I_{3R}$, $c_{i\alpha}^{R}, i \in I_2 \cup I_{3L}$, *and* $a_{j\alpha}^{L}$, $b_{j\alpha}^{R}$, $j = 1, \ldots, m$, are α-level optimal parameters for any M-α-Pareto optimal solution.

The relationships between the optimal solutions to (10.36) and the M-α-Pareto optimal concept of the Gα-MOLP can be characterized by the following theorems.

Theorem 10.6.4 (Minimax problem and M-α-Pareto optimality) *If* $\mathbf{x}^*$ *is a unique optimal solution to (10.36), then* $\mathbf{x}^*$ *is an M-α-Pareto optimal solution to the Gα-MOLP.*

Theorem 10.6.5 (M-α-Pareto optimality and minimax problem) *If* $\mathbf{x}^*$ *is an M-α-Pareto optimal solution to the Gα-MOLP, then* $\mathbf{x}^*$ *is an optimal solution to (10.36) for some* $\overline{\mu} = (\overline{\mu}_1, \ldots, \overline{\mu}_k)$.

The proofs of these theorems follow directly from the definitions of optimality and M-α-Pareto optimality by making use of contradiction arguments.

It must be observed here that for generating M-α-Pareto optimal solutions using Theorem 10.6.4, uniqueness of solution must be verified. In the ad hoc numeral approach, however, to test the M-α-Pareto optimality of a current optimal solution $\mathbf{x}^*$, we formulate and solve the following linear programming problem:

$$\left.\begin{array}{ll} \text{maximize} & \sum_{i=1}^{k} \varepsilon_i \\ \text{subject to} & \mathbf{c}_{i\alpha}^{L}\mathbf{x} + \varepsilon_i = \mathbf{c}_{i\alpha}^{L}\mathbf{x}^*, \quad \varepsilon_i \ge 0, \quad i \in I_1 \cup I_{3R} \\ & \mathbf{c}_{i\alpha}^{R}\mathbf{x} - \varepsilon_i = \mathbf{c}_{i\alpha}^{R}\mathbf{x}^*, \quad \varepsilon_i \ge 0, \quad i \in I_2 \cup I_{3L} \\ & \mathbf{a}_{j\alpha}^{L}\mathbf{x} \le b_{j\alpha}^{R}, \quad j = 1, \ldots, m, \quad \mathbf{x} \ge \mathbf{0} \end{array}\right\} \tag{10.40}$$

Let $\overline{\mathbf{x}}$ and $\overline{\varepsilon}_i$ be an optimal solution to this problem. If all $\overline{\varepsilon}_i = 0$, then $\mathbf{x}^*$ is an M-α-Pareto optimal solution. If at least one $\overline{\varepsilon}_i > 0$, as discussed previously, it can be easily shown that $\overline{\mathbf{x}}$ is an M-α-Pareto optimal solution.

Now given the M-α-Pareto optimal solution for the degree α and the reference membership values specified by the DM by solving the corresponding minimax problem, the DM must either be satisfied with the current M-α-Pareto optimal solution and α or update the reference membership values and/or the degree α. To help the DM express a degree of preference, trade-off information between a standing membership function and each of the other membership functions as well as between the degree α and the membership functions is very useful. Such trade-off information is easily obtainable since it is closely related to the simplex multipliers of the problem (10.38).

To derive the trade-off information, define the following Lagrangian function L corresponding to problem (10.38):

$$\begin{aligned} L \;=\; & \mathbf{c}_{1\alpha}^{L}\mathbf{x} + \sum_{i \in I_1 \cup I_{3R}} \pi_{iR}\{\mathbf{c}_{i\alpha}^{L}\mathbf{x} - d_{iR}^{-1}(\overline{\mu}_i - v^*)\} \\ & + \sum_{i \in I_2 \cup I_{3L}} \pi_{iL}\{d_{iL}^{-1}(\overline{\mu}_i - v^*) - \mathbf{c}_{i\alpha}^{R}\mathbf{x}\} + \sum_{j=1}^{m} \lambda_j(\mathbf{a}_{j\alpha}^{L}\mathbf{x} - b_{j\alpha}^{R}) \end{aligned} \tag{10.41}$$

where π_{iL}, π_{iR}, and λ_j are simplex multipliers corresponding to the constraints of (10.38).

Here we assume that problem (10.38) has a unique and nondegenerate optimal solution satisfying the following conditions:

(1) $\pi_{iR} > 0,\ i \in I_1 \cup I_{3R},\ i \neq 1$

(2) $\pi_{iL} > 0,\ i \in I_2 \cup I_{3L}$.

Then by using the results in Haimes and Chankong (1979), the following expression holds:

$$-\frac{\partial(\mathbf{c}_{1\alpha}^{L}\mathbf{x})}{\partial(\mathbf{c}_{i\alpha}^{L}\mathbf{x})} = \pi_{iR}, \quad i \in I_1 \cup I_{3R}, \quad i \neq 1 \tag{10.42}$$

$$-\frac{\partial(\mathbf{c}_{1\alpha}^{L}\mathbf{x})}{\partial(\mathbf{c}_{i\alpha}^{R}\mathbf{x})} = -\pi_{iL}, \quad i \in I_2 \cup I_{3L} \tag{10.43}$$

Furthermore, using the strictly monotone decreasing or increasing property of $d_{iR}(\cdot)$ or $d_{iL}(\cdot)$ together with the chain rule, if $d_{iR}(\cdot)$ and $d_{iL}(\cdot)$ are differentiable at the optimal solution to (10.38), it holds that

$$-\frac{\partial\mu_1(c_{1\alpha}^{L}\mathbf{x})}{\partial\mu_i(\mathbf{c}_{i\alpha}^{L}\mathbf{x})} = \frac{d_{1R}'(\mathbf{c}_{1\alpha}^{L}\mathbf{x})}{d_{iR}'(\mathbf{c}_{i\alpha}^{L}\mathbf{x})}\pi_{iR}, \quad i \in I_1 \cup I_{3R}, \quad i \neq 1 \tag{10.44}$$

$$-\frac{\partial\mu_1(\mathbf{c}_{1\alpha}^{L}\mathbf{x})}{\partial\mu_i(\mathbf{c}_{i\alpha}^{R}\mathbf{x})} = \frac{d_{1R}'(\mathbf{c}_{1\alpha}^{L}\mathbf{x})}{d_{iL}'(\mathbf{c}_{i\alpha}^{R}\mathbf{x})}\pi_{iL}, \quad i \in I_2 \cup I_{3L} \tag{10.45}$$

where $d'_{iR}(\cdot)$ and $d'_{iL}(\cdot)$ denote the differential coefficients of $d_{iR}(\cdot)$ and $d_{iL}(\cdot)$, respectively.

Regarding a trade-off rate between $\mu_1(\mathbf{c}^L_{1\alpha}\mathbf{x})$ and α, the following relation holds based on the sensitivity theorem (for details, see e.g., Fiacco (1983); Luenberger (1984)

$$\begin{aligned}\frac{\partial \mu_1(\mathbf{c}^L_{1\alpha}\mathbf{x})}{\partial \alpha} = {} & d'_{1R}(\mathbf{c}^L_{1\alpha}\mathbf{x})\left\{\frac{\partial(\mathbf{c}^L_{1\alpha})}{\partial \alpha}\mathbf{x} + \sum_{i\in I_1\cup I_{3R}} \pi_{iR}\frac{\partial(\mathbf{c}^L_{i\alpha})}{\partial \alpha}\mathbf{x}\right. \\ & \left. - \sum_{i\in I_2\cup I_{3L}} \pi_{iL}\frac{\partial(\mathbf{c}^R_{i\alpha})}{\partial \alpha}\mathbf{x} + \sum_{j=1}^{m}\lambda_j\{\frac{\partial(\mathbf{a}^L_{j\alpha})}{\partial \alpha}\mathbf{x} - \frac{\partial(b^R_{j\alpha})}{\partial \alpha}\}\right\} \end{aligned} \tag{10.46}$$

It should be noted that to obtain the trade-off rate information from (10.44) and (10.45), all the constraints of problem (10.38) must be active for the current optimal solution. Therefore, if there are inactive constraints, it is necessary to replace $\overline{\mu}_i$ for inactive constraints by $d_{iR}(\mathbf{c}^L_{i\alpha}\mathbf{x}^*) + v^*$ or $d_{iL}(\mathbf{c}^R_{i\alpha}\mathbf{x}^*) + v^*$ and solve the corresponding problem (10.38) for obtaining the simplex multipliers.

Now, following the above discussions, we can present the interactive algorithm to derive the satisficing solution for the DM from the M-α-Pareto optimal solution set. The steps marked with an asterisk involve interaction with the DM.

Interactive fuzzy multiobjective linear programming with fuzzy parameters

Step 0: (Individual minimum and maximum) Calculate the individual minimum and maximum of each objective function under the given constraints for $\alpha = 0$ and $\alpha = 1$.

Step 1*: (Membership functions) Elicit a membership function $\mu_i(\mathbf{c}_i\mathbf{x})$ from the DM for each of the objective functions.

Step 2*: (Initialization) Ask the DM to select the initial value of α ($0 \le \alpha \le 1$) and set the initial reference membership values $\overline{\mu}_i = 1$, $i = 1, \ldots, k$.

Step 3: (M-α-Pareto optimal solution) For the degree α and the reference membership values specified by the DM, solve the minimax problem and perform the M-α-Pareto optimality test to obtain the M-α-Pareto optimal solution and the trade-off rates between the membership functions and the degree α.

Step 4*: (Termination or updating) The DM is supplied with the corresponding M-α-Pareto optimal solution and the trade-off rates between the membership functions and the degree α. If the DM is satisfied with the current membership function values of the M-α-Pareto optimal solution and α, stop. Otherwise, the DM must update the reference membership values and/or the degree α by considering the current values of the membership functions and α together with the trade-off rates between the membership functions and the degree α and return to Step 3.

Here it should be stressed to the DM that (1) any improvement of one membership function can be achieved only at the expense of at least one of the other membership

functions for some fixed degree α and (2) the greater value of the degree α gives the worse values of the membership functions for some fixed reference membership values.

It is significant to point out here that all the results presented in this section have already been extended by the authors to deal with multiobjective linear fractional programming problems with fuzzy parameters. A successful generalization along this line can be found in Sakawa and Yano (1985a), and interested readers might refer to them for details.

10.7 Interactive fuzzy stochastic multiobjective linear programming

In this section, by considering the imprecision of a DM's judgments for stochastic objective functions and/or constraints in multiobjective linear programming problems, fuzzy stochastic multiobjective linear programming is introduced.

For that purpose, assuming that the coefficients in objective functions and right-hand side constants of constraints are random variables, we deal with stochastic multiobjective linear programming problems formulated as:

$$\left.\begin{array}{rr} \text{minimize} & z_1(\mathbf{x}, \omega) = \mathbf{c}_1(\omega)\mathbf{x} \\ & \vdots \\ \text{minimize} & z_k(\mathbf{x}, \omega) = \mathbf{c}_k(\omega)\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b}(\omega) \\ & \mathbf{x} \geq \mathbf{0} \end{array}\right\} \tag{10.47}$$

where $\mathbf{x}$ is an n dimensional decision variable column vector, and A is an $m \times n$ coefficient matrix.

It should be noted that $\mathbf{c}_l(\omega)$, $l = 1, \ldots, k$ are n dimensional random variable row vectors with finite mean $\bar{\mathbf{c}}_l$ and finite covariance matrix $V_l = (v^l_{jh}) = (\mathrm{Cov}\{c_{lj}(\omega), c_{lh}(\omega)\})$, $j = 1, \ldots, n$, $h = 1, \ldots, n$, and $b_i(\omega)$, $i = 1, \ldots, m$ are random variables with finite mean $\overline{b}_i$ which are independent of each other, and the distribution function of each of them is also assumed to be continuous and increasing.

Multiobjective linear programming problems with random variable coefficients are said to be stochastic multiobjective linear programming ones, which are often seen in actual decision making situations. For example, consider a production planning problem to optimize the gross profit and production cost simultaneously under the condition that unit profits of the products, unit production costs of them and maximal amounts of the resources depend on seasonal factors or market prices. Such a production planning problem can be formulated as a multiobjective programming problem with random variable coefficients expressed by (10.47).

Observing that the formulated problem (10.47) involves random variable coefficients, ordinary mathematical programming methods cannot be directly applied. Consequently, we deal with the constraints in (10.47) as chance constrained conditions (Charnes and Cooper 1959) which mean that the constraints need to be satisfied with a certain probability (satisficing level) and over. Namely, replacing the constraints in (10.47) by chance

constrained conditions with satisficing levels β_i, $i = 1, \ldots, m$, (10.47) can be converted as:

$$\left.\begin{array}{rr} \text{minimize} & z_1(\mathbf{x}, \omega) = \mathbf{c}_1(\omega)\mathbf{x} \\ & \vdots \\ \text{minimize} & z_k(\mathbf{x}, \omega) = \mathbf{c}_k(\omega)\mathbf{x} \\ \text{subject to} & \Pr[\mathbf{a}_1\mathbf{x} \le b_1(\omega)] \ge \beta_1 \\ & \vdots \\ & \Pr[\mathbf{a}_m\mathbf{x} \le b_m(\omega)] \ge \beta_m \\ & \mathbf{x} \ge \mathbf{0} \end{array}\right\} \tag{10.48}$$

where $\mathbf{a}_i$ is the ith row vector of A and $b_i(\omega)$ is the ith element of $\mathbf{b}(\omega)$.

Denoting continuous and increasing distribution functions of random variables $b_i(\omega)$, $i = 1, \ldots, m$ by $F_i(r) = \Pr[b_i(\omega) \le r]$, the ith constraint in (10.48) can be rewritten as:

$$\begin{aligned} \Pr[\mathbf{a}_i\mathbf{x} \le b_i(\omega)] \ge \beta_i &\Leftrightarrow 1 - \Pr[b_i(\omega) \le \mathbf{a}_i\mathbf{x}] \ge \beta_i \\ &\Leftrightarrow 1 - F_i(\mathbf{a}_i\mathbf{x}) \ge \beta_i \\ &\Leftrightarrow F_i(\mathbf{a}_i\mathbf{x}) \le 1 - \beta_i \\ &\Leftrightarrow \mathbf{a}_i\mathbf{x} \le F_i^{-1}(1 - \beta_i) \end{aligned} \tag{10.49}$$

Letting $\hat{b}_i = F_i^{-1}(1 - \beta_i)$, (10.48) can be transformed into the following equivalent problem:

$$\left.\begin{array}{rr} \text{minimize} & z_1(\mathbf{x}, \omega) = \mathbf{c}_1(\omega)\mathbf{x} \\ & \vdots \\ \text{minimize} & z_k(\mathbf{x}, \omega) = \mathbf{c}_k(\omega)\mathbf{x} \\ \text{subject to} & A\mathbf{x} \le \hat{\mathbf{b}} \\ & \mathbf{x} \ge \mathbf{0} \end{array}\right\} \tag{10.50}$$

where $\hat{\mathbf{b}} = (\hat{b}_1, \ldots, \hat{b}_m)^T$. In the following, for notational convenience, the feasible region of (10.50) is denoted by X.

For the multiobjective chance constrained programming problem (10.50), several stochastic models such as an expectation optimization model, a variance minimization model, a probability maximization model and a fractile criterion model, have been proposed depending on the concern of the DM.

In this section, due to space restrictions, we only introduce a probability maximization model for the multiobjective chance constrained programming problem (Sakawa and Kato 2002; Sakawa *et al*. 2004).

For details of the remaining models such as an expectation optimization model, a variance minimization model and a fractile criterion model, the readers might refer to the recently published book of Sakawa *et al*. (2011).

In a probability maximization model, the DM aims to maximize the probability that each objective function represented as a random variable is less than or equal to a certain permissible level in (10.50).

Substituting the minimization of the objective functions $z_l(\mathbf{x}, \omega) = \mathbf{c}_l(\omega)\mathbf{x}$, $l = 1, \ldots, k$ in (10.50) for the maximization of the probability that each of objective functions $z_l(\mathbf{x}, \omega)$ is less than or equal to a certain permissible level f_l, the problem can be converted as:

$$\left.\begin{array}{ll} \text{maximize} & p_1(\mathbf{x}) = \Pr\left[z_1(\mathbf{x}, \omega) \leq f_1\right] \\ & \vdots \\ \text{maximize} & p_k(\mathbf{x}) = \Pr\left[z_k(\mathbf{x}, \omega) \leq f_k\right] \\ \text{subject to} & A\mathbf{x} \leq \hat{\mathbf{b}} \\ & \mathbf{x} \geq \mathbf{0}. \end{array}\right\} \tag{10.51}$$

In order to consider the imprecise nature of the DM's judgment for each objective function in (10.51), if we introduce the fuzzy goals such as '$p_l(\mathbf{x})$ should be substantially greater than or equal to a certain value', (10.51) can be rewritten as:

$$\underset{\mathbf{x} \in X}{\text{maximize}}(\mu_1(p_1(\mathbf{x})), \ldots, \mu_k(p_k(\mathbf{x}))) \tag{10.52}$$

where $\mu_l(\cdot)$ is a membership function to quantify a fuzzy goal for the lth objective function in (10.51). To be more explicit, if the DM feels that $p_l(\mathbf{x})$ should be greater than or equal to at least $p_{l,0}$ and $p_l(\mathbf{x}) \geq p_{l,1}(> p_{l,0})$ is satisfactory, the shape of a typical membership function is as shown in Figure 10.6.

In an interactive fuzzy satisficing method, to generate a candidate for the satisficing solution which is also M-Pareto optimal, the DM is asked to specify the aspiration levels of achievement for the membership values of all membership functions, called the reference membership levels (Sakawa 1993; Sakawa and Yano 1985b, 1989, 1990a; Sakawa *et al*. 1987b).

For the DM's reference membership levels $\overline{\mu}_l$, $l = 1, \ldots, k$, the corresponding M-Pareto optimal solution, which is nearest to the requirements in the minimax sense or better than that if the reference membership levels are attainable, is obtained by solving

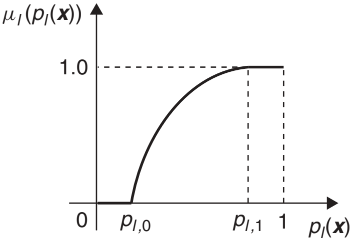

Figure 10.6 Example of a membership function $\mu_l(p_l(\mathbf{x}))$.

the following minimax problem

$$\left.\begin{array}{ll}\text{minimize} & \max\limits_{l=1,\ldots,k}\{\overline{\mu}_l - \mu_l(p_l(\mathbf{x}))\} \\ \text{subject to} & \mathbf{x} \in X.\end{array}\right\} \tag{10.53}$$

By introducing the auxiliary variable v, this problem can be equivalently transformed as:

$$\left.\begin{array}{ll}\text{minimize } v & \\ \text{subject to} & \overline{\mu}_1 - \mu_1(p_1(\mathbf{x})) \leq v \\ & \vdots \\ & \overline{\mu}_k - \mu_k(p_k(\mathbf{x})) \leq v \\ & \mathbf{x} \in X.\end{array}\right\} \tag{10.54}$$

Now, let every membership function $\mu_l(\cdot)$ be continuous and strictly increasing. Then, (10.54) is equivalent to the following problem.

$$\left.\begin{array}{ll}\text{minimize } v & \\ \text{subject to} & p_1(\mathbf{x}) \geq \mu_1^{-1}(\overline{\mu}_1 - v) \\ & \vdots \\ & p_k(\mathbf{x}) \geq \mu_k^{-1}(\overline{\mu}_k - v) \\ & \mathbf{x} \in X\end{array}\right\} \tag{10.55}$$

Since (10.55) is a nonconvex nonlinear programming problem in general, it is difficult to solve it.

Here, in (10.47), we assume that $\mathbf{c}_l(\omega)$, $l = 1, \ldots, k$ are n dimensional random variable row vectors expressed by $\mathbf{c}_l(\omega) = \mathbf{c}_l^1 + t_l(\omega)\mathbf{c}_l^2$ where $t_l(\omega)$'s are random variables independent of each other, and $\alpha_l(\omega)$'s are random variables expressed by $\alpha_l(\omega) = \alpha_l^1 + t_l(\omega)\alpha_l^2$, where the corresponding distribution function $T_l(\cdot)$ of each of $t_l(\omega)$s is assumed to be continuous and strictly increasing.

Supposing that $\mathbf{c}_l^2\mathbf{x} + \alpha_l^2 > 0$, $l = 1, \ldots, k$ for any $\mathbf{x} \in X$, from the assumption on distribution functions $T_l(\cdot)$ of random variables $t_l(\omega)$, we can rewrite the objective functions in (10.51) as:

$$\begin{aligned}\Pr\left[z_l(\mathbf{x}, \omega) \leq f_l\right] &= \Pr\left[(\mathbf{c}_l^1 + t_l(\omega)\mathbf{c}_l^2)\mathbf{x} + (\alpha_l^1 + t_l(\omega)\alpha_l^2) \leq f_l\right] \\ &= \Pr\left[(\mathbf{c}_l^2\mathbf{x} + \alpha_l^2)t_l(\omega) + (\mathbf{c}_l^1\mathbf{x} + \alpha_l^1) \leq f_l\right] \\ &= \Pr\left[t_l(\omega) \leq \frac{f_l - (\mathbf{c}_l^1\mathbf{x} + \alpha_l^1)}{(\mathbf{c}_l^2\mathbf{x} + \alpha_l^2)}\right]\end{aligned}$$

$$= T_l\left(\frac{f_l - \mathbf{c}_l^1\mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x} + \alpha_l^2}\right).$$

Hence, (10.51) can be transformed into the following ordinary multiobjective programming problem:

$$\left.\begin{aligned}&\text{maximize}\ \ p_1(\mathbf{x}) = T_1\left(\frac{f_1 - \mathbf{c}_1^1\mathbf{x} - \alpha_1^1}{\mathbf{c}_1^2\mathbf{x} + \alpha_1^2}\right)\\ &\qquad\qquad\vdots\\ &\text{maximize}\ \ p_k(\mathbf{x}) = T_k\left(\frac{f_k - \mathbf{c}_k^1\mathbf{x} - \alpha_k^1}{\mathbf{c}_k^2\mathbf{x} + \alpha_k^2}\right)\\ &\text{subject to}\ \ \mathbf{x} \in X\end{aligned}\right\} \tag{10.56}$$

In view of

$$p_l(\mathbf{x}) = T_l\left(\frac{f_l - \mathbf{c}_l^1\mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x} + \alpha_l^2}\right)$$

and the continuity and strictly increasing property of the distribution function $T_l(\cdot)$, this problem can be equivalently transformed as:

$$\left.\begin{aligned}&\text{minimize}\ \ v\\ &\text{subject to}\ \ \frac{f_1 - \mathbf{c}_1^1\mathbf{x} - \alpha_1^1}{\mathbf{c}_1^2\mathbf{x} + \alpha_1^2} \geq T_1^{-1}(\mu_1^{-1}(\overline{\mu}_1 - v))\\ &\qquad\qquad\vdots\\ &\qquad\qquad \frac{f_k - \mathbf{c}_k^1\mathbf{x} - \alpha_k^1}{\mathbf{c}_k^2\mathbf{x} + \alpha_k^2} \geq T_k^{-1}(\mu_k^{-1}(\overline{\mu}_k - v))\\ &\qquad\qquad \mathbf{x} \in X.\end{aligned}\right\} \tag{10.57}$$

It is important to note here that, in this formulation, if the value of v is fixed, it can be reduced to a set of linear inequalities. Obtaining the optimal solution v^* to the above problem is equivalent to determining the maximum value of v so that there exists an admissible set satisfying the constraints of Equation (10.57). Since v satisfies

$$\overline{\mu}_{\max} - \max_{l=1,\dots,k} \mu_{l,\max} \leq v \leq \overline{\mu}_{\max} - \min_{l=1,\dots,k} \mu_{l,\min}$$

where

$$\overline{\mu}_{\max} = \max_{l=1,\dots,k} \overline{\mu}_l,\ \ \mu_{l,\max} = \max_{\mathbf{x}\in X} \mu_l(p_l(\mathbf{x})),\ \ \mu_{l,\min} = \min_{\mathbf{x}\in X} \mu_l(p_l(\mathbf{x})),$$

we can obtain the minimum value of v by combined use of the bisection method and phase one of the linear programming technique.

After calculating v^*, the minimum value of v, we solve the following linear fractional programming problem in order to uniquely determine $\mathbf{x}^*$ corresponding to v^*.

$$\left.\begin{aligned} &\text{minimize} \quad \frac{\mathbf{c}_1^1\mathbf{x} + \alpha_1^1 - f_1}{\mathbf{c}_1^2\mathbf{x} + \alpha_1^2} \\ &\text{subject to} \quad \frac{f_2 - \mathbf{c}_2^1\mathbf{x} - \alpha_2^1}{\mathbf{c}_2^2\mathbf{x} + \alpha_2^2} \geq T_2^{-1}(\mu_2^{-1}(\overline{\mu}_2 - v^*)) \\ &\qquad\qquad \vdots \\ &\qquad\qquad \frac{f_k - \mathbf{c}_k^1\mathbf{x} - \alpha_k^1}{\mathbf{c}_k^2\mathbf{x} + \alpha_k^2} \geq T_k^{-1}(\mu_k^{-1}(\overline{\mu}_k - v^*)) \\ &\qquad\qquad \mathbf{x} \in X \end{aligned}\right\} \tag{10.58}$$

where $z_1(\mathbf{x}, \omega)$ is supposed to be the most important to the DM.

Using the Charnes – Cooper transformation (Charnes and Cooper 1962)

$$s = 1/(\mathbf{c}_1^2\mathbf{x} + \alpha_1^2), \quad \mathbf{y} = s \cdot \mathbf{x}, \quad s > 0, \tag{10.59}$$

the linear fractional programming problem (10.58) is converted to the following linear programming problem

$$\left.\begin{aligned} \text{minimize} \qquad & \mathbf{c}_1^1\mathbf{y} + (\alpha_1^1 - f_1) \cdot s \\ \text{subject to} \quad & \tau_2 \cdot (\mathbf{c}_2^2\mathbf{y} + \alpha_2^2 \cdot s) + \mathbf{c}_2^1\mathbf{y} + (\alpha_2^1 - f_2) \cdot s \leq 0 \\ & \vdots \\ & \tau_k \cdot (\mathbf{c}_k^2\mathbf{y} + \alpha_k^2 \cdot s) + \mathbf{c}_k^1\mathbf{y} + (\alpha_k^1 - f_k) \cdot s \leq 0 \\ & A\mathbf{y} - s \cdot \hat{\mathbf{b}} \leq \mathbf{0} \\ & \mathbf{c}_1^2\mathbf{y} + \alpha_1^2 \cdot s = 1 \\ & -s \leq -\delta \\ & \mathbf{y} \geq \mathbf{0} \\ & t \geq 0 \end{aligned}\right\} \tag{10.60}$$

where $\tau_l = T_l^{-1}(\mu_l^{-1}(\overline{\mu}_2 - v^*))$, and δ is sufficiently small and positive. If the optimal solution $(\mathbf{y}^*, s^*)$ to (10.60) is not unique, the Pareto optimality of $\mathbf{x}^* = s^* \cdot \mathbf{y}^*$ is not guaranteed. The Pareto optimality of $\mathbf{x}^*$ can be tested by solving the following linear programming problem.

$$\left.\begin{aligned} &\text{maximize}\ w = \sum_{l=1}^{k} \varepsilon_l \\ &\text{subject to}\ q_1(\mathbf{x}) - \varepsilon_1 = \frac{q_1(\mathbf{x}^*)}{r_1(\mathbf{x}^*)} \cdot r_1(\mathbf{x}) \\ &\qquad\qquad \vdots \\ &\qquad q_k(\mathbf{x}) - \varepsilon_k = \frac{q_k(\mathbf{x}^*)}{r_k(\mathbf{x}^*)} \cdot r_k(\mathbf{x}) \\ &\qquad \mathbf{x} \in X,\ \ \boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_k)^T \geq \mathbf{0} \end{aligned}\right\} \tag{10.61}$$

where

$$q_l(\mathbf{x}) = f_l - \mathbf{c}_l^1 \mathbf{x} - \alpha_l^1,\ \ r_l(\mathbf{x}) = \mathbf{c}_l^2 \mathbf{x} + \alpha_l^2.$$

For the optimal solution to (10.61), if $w = 0$, i.e., $\varepsilon_l = 0$ for $l = 1, \ldots, k$, $\mathbf{x}^*$ is Pareto optimal. On the other hand, if $w > 0$, i.e., $\varepsilon_l > 0$ for at least one l, $\mathbf{x}^*$ is not Pareto optimal. Then, we can find a Pareto optimal solution according to the following algorithm.

Step 1: For the optimal solution $\overline{\mathbf{x}}, \overline{\boldsymbol{\varepsilon}}$ to the problem (10.61), after arbitrarily selecting j such as $\overline{\varepsilon}_j > 0$, solve the following problem.

$$\left.\begin{aligned} &\text{maximize} \frac{f_j - \mathbf{c}_j^1 \mathbf{x} - \alpha_j^1}{\mathbf{c}_j^2 \mathbf{x} + \alpha_j^2} \\ &\text{subject to} \frac{f_l - \mathbf{c}_l^1 \mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2 \mathbf{x} + \alpha_l^2} = \frac{f_l - \mathbf{c}_l^1 \overline{\mathbf{x}} - \alpha_l^1}{\mathbf{c}_l^2 \overline{\mathbf{x}} + \alpha_l^2},\ \ \{l \mid \overline{\varepsilon}_l = 0\} \\ &\qquad \frac{f_l - \mathbf{c}_l^1 \mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2 \mathbf{x} + \alpha_l^2} \geq \frac{f_l - \mathbf{c}_l^1 \overline{\mathbf{x}} - \alpha_l^1}{\mathbf{c}_l^2 \overline{\mathbf{x}} + \alpha_l^2},\ \ \{l \mid \overline{\varepsilon}_l > 0\} \\ &\qquad \mathbf{x} \in X \end{aligned}\right\} \tag{10.62}$$

Since the above problem can be converted to a linear programming problem by the Charnes–Cooper transformation (Charnes and Cooper 1962), we can solve it by the simplex method.

Step 2: To test the Pareto optimality of the optimal solution $\hat{\mathbf{x}}$ to (10.62), solve the problem (10.61) where $\hat{\mathbf{x}}$ is substituted for $\mathbf{x}^*$.

Step 3: If $w = 0$, $\hat{\mathbf{x}}$ is Pareto optimal and stop. Otherwise, i.e., if $w > 0$, return to Step 1 since $\hat{\mathbf{x}}$ is not Pareto optimal.

Repeating this process at least $k-1$ iterations, a Pareto optimal solution can be obtained.

The DM must either be satisfied with the current Pareto optimal solution or act on this solution by updating the reference membership levels. In order to help the DM express a degree of preference, trade-off information between a standing membership function and each of the other membership functions is very useful. Such trade-off information is easily obtainable since it is closely related to the simplex multipliers of (10.60).

To derive the trade-off information, define the Lagrange function L for (10.60) as follows:

$$\begin{aligned} L(\mathbf{y}, s, \pi, \sigma, \omega) = {} & \mathbf{c}_1^1\mathbf{y} + (\alpha_1^1 - f_1)\cdot s + \sum_{l=2}^{k} \pi_l[\tau_l\cdot(\mathbf{c}_l^2\mathbf{y} + \alpha_l^2\cdot s) \\ & + \{\mathbf{c}_l^1\mathbf{y} + (\alpha_l^1 - f_l)\cdot s\}] + \sum_{i=1}^{m} \sigma_i(\mathbf{a}_i\mathbf{y} - s\cdot\mathbf{b}) \\ & + \sigma_{m+1}\cdot(\mathbf{c}_1^2\mathbf{y} + \alpha_1^2\cdot s - 1) + \sigma_{m+2}\cdot(-s+\delta) \\ & - \sum_{j=1}^{n} \omega_j\cdot y_j - \omega_{n+1}\cdot s \end{aligned} \tag{10.63}$$

where π, σ and ω are simplex multipliers.

Then, the partial derivative of $L(\mathbf{y}, s, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\omega})$ with respect to τ_l is given as follows:

$$\frac{\partial L(\mathbf{y}, s, \boldsymbol{\pi}, \boldsymbol{\sigma}, \boldsymbol{\omega})}{\partial \tau_l} = \pi_l\cdot(\mathbf{c}_l^2\mathbf{y} + \alpha_l^2\cdot s), \ \ l = 2, \ldots, k \tag{10.64}$$

On the other hand, for the optimal solution $(\mathbf{y}^*, s^*)$ to (10.60) and the corresponding simplex multipliers $(\boldsymbol{\pi}^*, \boldsymbol{\sigma}^*, \boldsymbol{\omega}^*)$, the following equation holds from the Kuhn–Tucker necessity theorem (Sakawa 1993).

$$L(\mathbf{y}^*, s^*, \boldsymbol{\pi}^*, \boldsymbol{\sigma}^*, \boldsymbol{\omega}^*) = \mathbf{c}_1^1\mathbf{y}^* + (\alpha_1^1 - f_1)\cdot s^* \tag{10.65}$$

If the first $(k-1)$ constraints to (10.60) are active, τ_l is calculated as follows:

$$\tau_l = -\frac{\mathbf{c}_l^1\mathbf{y}^* + (\alpha_l^1 - f_l)\cdot s^*}{\mathbf{c}_l^2\mathbf{y}^* + \alpha_l^2\cdot s^*}, \ \ l = 2, \ldots, k. \tag{10.66}$$

From (10.64), (10.65) and (10.66), for $l = 2, \ldots, k$, we have

$$\frac{\partial(\mathbf{c}_1^1\mathbf{y}^* + (\alpha_1^1 - f_1)\cdot s^*)}{\partial\left(-\dfrac{\mathbf{c}_l^1\mathbf{y}^* + (\alpha_l^1 - f_l)\cdot s^*}{\mathbf{c}_l^2\mathbf{y}^* + \alpha_l^2\cdot s^*}\right)} = \pi_l^*\cdot(\mathbf{c}_l^2\mathbf{y}^* + \alpha_l^2\cdot s^*). \tag{10.67}$$

By substituting $\mathbf{x}^*$ for $\mathbf{y}^*$, s^* in (10.67), the equation is rewritten as

$$-\frac{\partial\left(\dfrac{f_1 - \mathbf{c}_1^1\mathbf{x}^* - \alpha_1^1}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2}\right)}{\partial\left(\dfrac{f_l - \mathbf{c}_l^1\mathbf{x}^* - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}\right)} = \pi_l^*\cdot\frac{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2}, \ \ l = 2, \ldots, k. \tag{10.68}$$

Using the chain rule, the following relation holds:

$$-\frac{\partial T_1\left(\dfrac{f_1 - \mathbf{c}_1^1\mathbf{x}^* - \alpha_1^1}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2}\right)}{\partial T_l\left(\dfrac{f_l - \mathbf{c}_l^1\mathbf{x}^* - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}\right)} = \pi_l^* \cdot \frac{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2} \cdot \frac{T_1^{'}\left(\dfrac{f_l - \mathbf{c}_1^1\mathbf{x}^* - \alpha_1^1}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2}\right)}{T_l^{'}\left(\dfrac{f_l - \mathbf{c}_l^1\mathbf{x}^* - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}\right)},$$

$$l = 2, \ldots, k. \tag{10.69}$$

Equivalently,

$$-\frac{\partial p_1(\mathbf{x}^*)}{\partial p_l(\mathbf{x}^*)} = \pi_l^* \cdot \frac{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2} \cdot \frac{p_1^{'}(\mathbf{x}^*)}{p_l^{'}(\mathbf{x}^*)}, \quad l = 2, \ldots, k. \tag{10.70}$$

Again, using the chain rule, for $l = 2, \ldots, k$, we have

$$-\frac{\partial \mu_1(p_1(\mathbf{x}^*))}{\partial \mu_l(p_l(\mathbf{x}^*))} = \pi_l^* \cdot \frac{\mathbf{c}_l^2\mathbf{x}^* + \alpha_l^2}{\mathbf{c}_1^2\mathbf{x}^* + \alpha_1^2} \cdot \frac{p_1^{'}(\mathbf{x}^*)}{p_l^{'}(\mathbf{x}^*)} \cdot \frac{\mu_1^{'}(p_1(\mathbf{x}^*))}{\mu_l^{'}(p_l(\mathbf{x}^*))}. \tag{10.71}$$

It should be stressed here that in order to obtain the trade-off information from (10.71), the first $(k - 1)$ constraints in (10.60) must be active. Therefore, if there are inactive constraints, it is necessary to replace $\overline{\mu}_l$ for inactive constraints with $\mu_l(p_l(\mathbf{x}^*)) + v^*$ and solve the corresponding problem to obtain the simplex multipliers.

Following the preceding discussions, we can now construct the interactive algorithm in order to derive the satisficing solution for the DM from the Pareto optimal solution set.

Interactive fuzzy stochastic multiobjective linear programming through probability maximization

Step 1: Calculating the individual minimum $\overline{z}_l^{\min}$ and maximum $\overline{z}_l^{\max}$ of $\mathrm{E}[z_l(\mathbf{x}, \omega)] = \overline{z}_l(\mathbf{x})$, $l = 1, \ldots, k$ under the chance constrained conditions with satisficing levels β_l, $l = 1, \ldots, m$.

Step 2: Ask the DM to specify permissible levels f_l, $l = 1, \ldots, k$ for objective functions.

Step 3: Calculate the individual minimum $p_{l,\min}$ and maximum $p_{l,\max}$ of $p_l(\mathbf{x})$, $l = 1, \ldots, k$ in the multiobjective probability maximization problem (10.56) by solving the following problems:

$$\underset{\mathbf{x}\in X}{\text{minimize}}\; p_l(\mathbf{x}) = T_l\left(\frac{f_l - \mathbf{c}_l^1\mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x} + \alpha_l^2}\right), \; l = 1, \ldots, k, \tag{10.72}$$

$$\underset{\mathbf{x}\in X}{\text{maximize}}\; p_l(\mathbf{x}) = T_l\left(\frac{f_l - \mathbf{c}_l^1\mathbf{x} - \alpha_l^1}{\mathbf{c}_l^2\mathbf{x} + \alpha_l^2}\right), \; l = 1, \ldots, k. \tag{10.73}$$

Then ask the DM to determine membership functions $\mu_l(p_l(\mathbf{x}))$ for objective functions in (10.56).

Step 4: Ask the DM to set the initial reference membership levels $\overline{\mu}_l = 1$, $l = 1, \ldots, k$.

Step 5: In order to obtain the optimal solution $\mathbf{x}^*$ to the minimax problem (10.53) corresponding to the reference membership levels $\overline{\mu}_l$, $l = 1, \ldots, k$, after solving (10.57) by the bisection method and phase one of the two-phase simplex method, solve the linear programming problem (10.60). For the obtained $\mathbf{x}^*$, if there are inactive constraints in the first $(k-1)$ constraints, replace $\overline{\mu}_l$ for inactive constraints with $\mu_l(p_l(\mathbf{x}^*)) + v^*$ and resolve the corresponding problem. Furthermore, if the obtained $\mathbf{x}^*$ is not unique, perform the Pareto optimality test.

Step 6: The DM is supplied with the corresponding Pareto optimal solution and the trade-off rates between the membership functions. If the DM is satisfied with the current membership function values of the Pareto optimal solution, stop. Otherwise, ask the DM to update the reference membership levels $\overline{\mu}_l$, $l = 1, \ldots, k$ by considering the current membership function values $\mu_l(p_l(\mathbf{x}^*))$ together with the trade-off rates $-\partial\mu_1/\partial\mu_l$, $l = 2, \ldots, k$, and return to Step 5.

Since the trade-off rates $-\partial\mu_1/\partial\mu_l$, $l = 2, \ldots, k$ in Step 6 indicate the decrement of value of a membership function μ_1 with a unit increment of value of a membership function μ_l, they are employed to estimate the local shape of $(\mu_1(p_1(\mathbf{x}^*)), \ldots, \mu_k(p_k(\mathbf{x}^*)))$ around $\mathbf{x}^*$.

Here, as in the discussion for fuzzy multiobjective linear programming, it should be also stressed to the DM that any improvement of one membership function can be achieved only at the expense of at least one of the other membership functions.

10.8 Related works and applications

So far multiobjective linear programming in fuzzy and stochastic environments has been briefly discussed on the basis of the author's continuing research work.

For further details of multiobjective linear and nonlinear programming in a fuzzy environment, including interactive computer programs and some applications, the readers might refer to Sakawa's book entitled 'Fuzzy Sets and Interactive Multiobjective Optimization' (Sakawa 1993). In addition, the books by Lai and Hwang (1994), Sakawa (2000, 2001), and Carlsson and Fullér (2002) together with the edited volumes of Kacprzyk and Orlovski (1987), Verdegay and Delgado (1989), Slowinski and Teghem (1990), Delgado *et al*. (1994), Slowinski (1998), Ehrgott and Gandibleux (2002), and Kahraman (2008) would be very useful for interested readers.

Extensions to two-level linear programming problems, where the upper level DM makes a decision first and the lower level DM makes a decision after understanding the decision of the upper level DM, can be found in the books by Sakawa and Nishizaki (2009), and Sakawa *et al*. (2011). Especially, Sakawa *et al*. (2011) covers the major aspects of the so-called fuzzy stochastic multiobjective programming.

Finally, it is appropriate to mention some application aspects of multiobjective optimization in fuzzy and stochastic environments. As we look at engineering, industrial,

and management applications of fuzzy multiobjective optimization, we can see continuing advances. They can be found, for example, in the areas of an air pollution regulation problem (Sommer and Pollatschek 1978), media selection in advertising (Wiedey and Zimmermann 1978), a transportation problem (Verdegay 1984), environmental planning (Sakawa and Yano 1985b), water supply system development planning (Slowinski 1986), operation of a packaging system in automated warehouses (Sakawa *et al*. 1987b), pass scheduling for hot tandem mills (Sakawa *et al*. 1987a), spatial planning problems (Leung 1988), profit apportionment in concerns (Ostermark 1988), a capital asset pricing model Ostermark (1989), a farm structure optimization problem (Czyzak 1990), diet optimization problems (Czyzak and Slowinski 1990), a forest management problem (Pickens and Hof 1991), quality control (Chakraborty 1994), wastewater management (Duckstein *et al*. 1994), fuzzy vehicle routing and scheduling (Cheng and Gen 1996), flexible scheduling in a machining center (Sakawa *et al*. 1996), a real size manpower allocation problem (Abboud *et al*. 1998), multiobjective interval transportation problems (Das *et al*. 1999), coal purchase planning in electric power plants (Shiroumaru *et al*. 2000), fuzzy job shop scheduling (Sakawa and Kubota 2000), and profit and cost allocation for a production, transportation problem (Sakawa *et al*. 2001b). Recent applications of fuzzy multiobjective optimization include fire station locations (Yang *et al*. 2007), no-wait flow shop sheduling (Javadi *et al*. 2008), integrated multi-product and multi-time period production/distribution planning decisions (Liang 2008), crop area planning (Zeng *et al*. 2010), environmental supply chain network design (Pishvaee and Razmi 2012), and operation planning in district heating and cooling plants (Sakawa and Matsui 2011).

References

Abboud N, Inuiguchi M, Sakawa M and Uemura Y (1998) Manpower allocation using genetic annealing. *European Journal of Operational Research* **111**, 405–420.

Bellman RE and Zadeh LA (1970) Decision making in a fuzzy environment. *Management Science* **17**, 141–164.

Benayoun R, de Montgofier J, Tergny J and Larichev O (1971) Linear programming with multiple objective functions, Step method (STEM). *Mathematical Programming* **1**, 366–375.

Carlsson C and Fullér R (2002) *Fuzzy Reasoning in Decision Making and Optimization*. Physica-Verlag, Heidelberg.

Chakraborty TK (1994) A class of single sampling inspection plans based on possibilistic programming problems. *Fuzzy Sets and Systems* **63**, 35–43.

Chankong V and Haimes YY (1983) *Multiobjective Decision Making: Theory and Methodology*. North-Holland, Amsterdam.

Charnes A and Cooper WW (1959) Chance constrained programming. *Management Science* **6**, 73–79.

Charnes A and Cooper WW (1962) Programming with linear fractional functions. *Naval Research Logistics Quarterly* **9**, 181–186.

Cheng R and Gen M (1996) Fuzzy vehicle routing and scheduling problem using genetic algorithms. In *Genetic Algorithms and Soft Computing* (eds Herrera F and Verdegay JL). Physica-Verlag, Heidelberg, pp. 683–709.

Choo EU and Atkins DR (1980) An interactive algorithm for multicriteria programming. *Computers and Operations Research* **7**, 81–87.

Czyzak P (1990) Application of the 'FLIP' method to farm structure optimization under uncertainty. In *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming Problems under Uncertainty* (eds Slowinski R and Teghem J). Kluwer Academic, Dordrecht, pp. 263–278.

Czyzak P and Slowinski R (1990) Solving multiobjective diet optimization problems under uncertainty. In *Multiple Criteria Decision Support* (eds Korhonen P, Lewandowski A and Wallenius J). Springer-Verlag, Berlin, pp. 272–281.

Dantzig GB (1955) Linear programming under uncertainty. *Management Science* **1**, 197–206.

Dantzig GB (1961) *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ.

Das SK, Goswami A and Alam SS (1999) Multiobjective transportation problem with interval cost, source and destination parameters. *European Journal of Operational Research* **117**, 100–112.

Delgado M, Kacprzyk J, Verdegay JL and Vila MA (eds) (1994) *Fuzzy Optimization: Recent Advances*. Physica-Verlag, Heidelberg.

Dubois D and Prade H (1978) Operations on fuzzy numbers. *International Journal of Systems Science* **9**, 613–626.

Dubois D and Prade H (1980) *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York.

Duckstein L, Bardossy A, Tecle A and Bogardi I (1994) Fuzzy composite programming with application to wastewater management under changing physical conditions. In *Fuzzy Optimization: Recent Advances* (eds Delgado M, Kacprzyk J, Verdegay JL and Vila MA). Physica-Verlag, Heidelberg, pp. 199–219.

Ehrgott M and Gandibleux X (eds) (2002) *Multiple Criteria Optimization – State of the Art Annotated Bibliographic Surveys*. Kluwer Academic, Boston, MA.

Fiacco AV (1983) *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York, NY.

Fichefet J (1976) GPSTEM: an interactive multiobjective optimization method. In *Progress in Operations Research* (ed. Prekopa A), vol. 1. North-Holland, Amsterdam, pp. 317–332.

Haimes YY and Chankong V (1979) Kuhn-Tucker multipliers as trade-offs in multiobjective decision-making analysis. *Automatica* **15**, 59–72.

Hannan EL (1981) Linear programming with multiple fuzzy goals. *Fuzzy Sets and Systems* **6**, 235–248.

Hulsurkar S, Biswal MP and Sinha SB (1997) Fuzzy programming approach to multi-objective stochastic linear programming problems. *Fuzzy Sets and Systems* **88**, 173–181.

Javadi B, Saidi-Mehrabad M, Haji A, Mahdavi I, Jolai F and Mahdavi-Amiri N (2008) No-wait flow shop sheduling using fuzzy multi-objective linear programming. *Journal of the Franklin Institute* **345**, 452–467.

Kacprzyk J and Orlovski SA (eds) (1987) *Optimization Models Using Fuzzy Sets and Possibility Theory*. D. Reidel Publishing Company, Dordrecht.

Kahraman C (ed) (2008) *Fuzzy Multi-Criteria Decision Making–Theory and Applications with Recent Developments*. Springer, New York, NY.

Köksalan M, Wallenius J and Zionts S (eds) (2011) *Multiple Criteria Decision Making – From Early History to the 21st Century*. World Scientific, Hackensack, NJ.

Lai YJ and Hwang CL (1994) *Fuzzy Multiple Objective Decision Making: Methods and Applications.* Springer-Verlag, Berlin.

Leberling H (1981) On finding compromise solution in multicriteria problems using the fuzzy min-operator. *Fuzzy Sets and Systems* **6**, 105–118.

Leclercq JP (1982) Stochastic programming: an interactive multicriteria approach. *European Journal of Operational Research* **10**, 33–41.

Leung Y (1988) *Spatial Analysis and Planning under Imprecision*. North-Holland, Amsterdam.

Liang TF (2008) Fuzzy multi-objective production/distribution planning decisions with multi-product and multi-time period in a supply chain. *Computers and Industrial Engineering* **55**, 676–694.

Luenberger DG (1984) *Linear and Nonlinear Programming*, 2nd edn. Addison-Wesley, San Francisco, CA.

Luhandjula MK (1987) Multiple objective programming problems with possibilistic coefficients. *Fuzzy Sets and Systems* **21**, 135–145.

Ostermark R (1988) Profit apportionment in concerns with mutual ownership: An application of fuzzy inequalities. *Fuzzy Sets and Systems* **26**, 283–297.

Ostermark R (1989) Fuzzy linear constraints in the capital asset pricing model. *Fuzzy Sets and Systems* **30**, 93–102.

Pickens JB and Hof JG (1991) Fuzzy goal programming in forestry: An application with special solution problems. *Fuzz* **39**, 239–246.

Pishvaee MS and Razmi J (2012) Environmental supply chain network design using multi-objective fuzzy mathematical programming. *Applied Mathematical Modelling* **36**, 3433–3446.

Rommelfanger H (1996) Fuzzy linear programming and applications. *European Journal of Operational Research* **92**, 512–527.

Sakawa M (1983) Interactive computer programs for fuzzy linear programming with multiple objectives. *International Journal of Man-Machine Studies* **18**, 489–503.

Sakawa M (1993) *Fuzzy Sets and Interactive Multiobjective Optimization*. Plenum Press, New York.

Sakawa M (2000) *Large Scale Interactive Fuzzy Multiobjective Optimization*. Physica-Verlag, Heidelberg.

Sakawa M (2001) *Genetic Algorithms and Fuzzy Multiobjective Optimization*. Kluwer Academic, Boston, MA.

Sakawa M (2002) Fuzzy multiobjective and multilevel optimization. In *Multiple Criteria Optimization – State of the Art Annotated Bibliographic Surveys* (eds Ehrgott M and Gandibleux X). Kluwer Academic, Boston, MA, pp. 171–226.

Sakawa M, Katagiri H and Kato K (2001a) An interactive fuzzy satisficing method for multiobjective stochastic linear programming problems using a fractile criterion model. *The 10th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 948–951.

Sakawa M and Kato K (2002) An interactive fuzzy satisficing method for multiobjective stochastic linear programming problems using chance constrained conditions. *Journal of Multi-Criteria Decision Analysis* **11**, 125–137.

Sakawa M and Kato K (2008) Interactive fuzzy multi-objective stochastic linear programming. In *Fuzzy Multi-Criteria Decision Making – Theory and Applications with Recent Developments* (ed. Kahraman C). Springer, New York, NY, pp. 375–408.

Sakawa M, Kato K and Katagiri H (2002) An interactive fuzzy satisficing method through a variance minimization model for multiobjective linear programming problems involving random variables. In *Knowledge-based Intelligent Information Engineering Systems & Allied Technologies (KES2002)* (eds Damiani E, Howlett RJ, Jain LC and Ichalkaranje), vol. 2. IOS Press, Amsterdam, pp. 1222–1226.

Sakawa M, Kato K and Katagiri H (2004) An interactive fuzzy satisficing method for multiobjective linear programming problems with random variable coefficients through a probability maximization model. *Fuzzy Sets and Systems* **146**, 205–220.

Sakawa M, Kato K and Mori T (1996) Flexible scheduling in a machining center through genetic algorithms. *Computers and Industrial Engineering* **30**, 931–940.

Sakawa M, Kato K and Nishizaki I (2003) An interactive fuzzy satisficing method for multiobjective stochastic linear programming problems through an expectation model. *European Journal of Operational Research* **144**, 581–597.

Sakawa M and Kubota R (2000) Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of Operational Research* **120**, 393–407.

Sakawa M and Matsui T (2011) Fuzzy multiobjective nonlinear operation planning in district heating and cooling plants. *Fuzzy Sets and Systems* doi:10.1016/j.fss.2011.10.020.

Sakawa M, Narazaki H, Konishi M, Nose K and Morita T (1987a) A fuzzy satisficing approach to multiobjective pass scheduling for hot tandem mills. In *Toward Interactive and Intelligent Decision Support Systems* (eds Sawaragi Y, Inoue K and Nakayama H), vol. 1. Springer-Verlag, Berlin, pp. 363–373.

Sakawa M and Nishizaki I (2009) *Cooperative and Noncooperative Multi-level Programming*. Springer, New York.

Sakawa M, Nishizaki I and Katagiri H (2011) *Fuzzy Stochastic Multiobjective Programming*. Springer, New York.

Sakawa M, Nishizaki I and Uemura Y (2001b) Fuzzy programming and profit and cost allocation for a production and transportation problem. *European Journal of Operational Research* **131**, 1–15.

Sakawa M and Yano H (1985a) Interactive decision making for multiobjective linear fractional programming problems with fuzzy parameters. *Cybernetics and Systems* **16**, 377–394.

Sakawa M and Yano H (1985b) An interactive fuzzy satisficing method using augmented minimax problems and its application to environmental systems. *IEEE Transactions on Systems Man and Cybernetics* **15**, 720–729.

Sakawa M and Yano H (1986) Interactive decision making for multiobjective linear problems with fuzzy parameters. In *Large-Scale Modeling and Interactive Decision Analysis* (eds Fandel G, Grauer M, Kurzhanski A and Wierzbicki AP). Springer-Verlag, New York, NY, pp. 88–96.

Sakawa M and Yano H (1989) Interactive decision making for multiobjective nonlinear programming problems with fuzzy parameters. *Fuzzy Sets and Systems* **29**, 129–144.

Sakawa M and Yano H (1990a) An interactive fuzzy satisficing method for generalized multiobjective linear programming problems with fuzzy parameters. *Fuzzy Sets and Systems* **35**, 125–142.

Sakawa M and Yano H (1990b) Trade-off rates in the hyperplane method for multiobjective optimization problems. *European Journal of Operational Research* **44**, 105–118.

Sakawa M, Yano H and Yumine T (1987b) An interactive fuzzy satisficing method for multiobjective linear-programming problems and its application. *IEEE Transactions on Systems Man and Cybernetics* **17**, 654–661.

Sakawa M, Yumine T and Yano H (1984) An interactive fuzzy satisficing method for multiobjective nonlinear programming problems. Technical Report CP-84-18, International Institute for Applied Systems Analysis, Laxenburg, Austria.

Shiroumaru I, Inuiguchi M and Sakawa M (2000) A fuzzy satisficing method for electric power plant coal purchase using genetic algorithms. *European Journal of Operational Research* **126**, 218–230.

Slowinski R (1986) A multicriteria fuzzy linear programming method for water supply system development planning. *Fuzzy Sets and Systems* **19**, 217–237.

Slowinski R (ed) (1998) *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*. Kluwer Academic, Boston, MA.

Slowinski R and Teghem J (eds) (1990) *Stochastic versus Fuzzy Approaches to Multiobjective Mathematical Programming Problems under Uncertainty*. Kluwer Academic, Dordrecht.

Sommer G and Pollatschek MA (1978) A fuzzy programming approach to an air pollution regulation problem. In *Progress in Cybernetics and Systems Research* (eds Trappl R, Klir GJ and Ricciardi L). Hemisphere, Washington, DC.

Stancu-Minasian IM (1984) *Stochastic Programming with Multiple Objective Functions*. D. Reidel Publishing Company, Dordrecht.

Stancu-Minasian IM (1990) Overview of different approaches for solving stochastic programming problems with multiple objective functions. In *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty* (eds Slowinski R and Teghem J). Kluwer Academic, Dordrecht, pp. 71–101.

Steuer RE (1986) *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, Ltd, New York, NY.

Steuer RE and Choo EU (1983) An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming* **26**, 326–344.

Teghem J, Dufrane D, Thauvoye M and P. K (1986) STRANGE: an interactive method for multiobjective linear programming under uncertainty. *European Journal of Operational Research* **26**, 65–82.

Verdegay JL (1984) Application of fuzzy optimization in operational research. *Control and Cybernetics* **13**, 229–239.

Verdegay JL and Delgado M (eds) (1989) *The Interface between Artificial Intelligence and Operations Research in Fuzzy Environment*. Verlag TÜV Rheinland, Cologne.

Wiedey G and Zimmermann HJ (1978) Media selection and fuzzy linear programming. *Journal of the Operational Research Society* **29**, 1071–1084.

Wierzbicki AP (1980) The use of reference objectives in multiobjective optimization. In *Multiple Criteria Decision Making: Theory and Application* (eds Fandel G and Gal T). Springer-Verlag, Berlin, pp. 468–486.

Yang L, Jones BF and Yang SH (2007) A fuzzy multi-objective programming for optimization of fire station locations through genetic algorithms. *European Journal of Operational Research* **181**, 903–915.

Zeng X, Kang S, Li F, Zhang L and Guo P (2010) Fuzzy multi-objective linear programming applying to crop area planning. *Agricultural Water Management* **98**, 134–142.

Zimmermann HJ (1976) Description and optimization of fuzzy systems. *International Journal of General Systems* **2**, 209–215.

Zimmermann HJ (1978) Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems* **1**, 45–55.

Zimmermann HJ (1983) Fuzzy mathematical programming. *Computers and Operations Research* **10**, 291–298.

Zimmermann HJ (1987) *Fuzzy Sets, Decision-Making and Expert Systems*. Kluwer Academic, Boston, MA.

Part V

APPLICATIONS IN MANAGEMENT AND ENGINEERING

11

Multiple criteria decision aid and agents: Supporting effective resource federation in virtual organizations

Pavlos Delias[1] and Nikolaos Matsatsinis[2]
[1]*Department of Accountancy, Kavala Institute of Technology, Kavala, Greece*
[2]*Department of Production Engineering and Management, Technical University of Crete, Greece*

11.1 Introduction

A virtual organization (VO) is a linkage of organizations, which share some of their resources to achieve their individual goals. A common instantiation of this practice is the case of grid computing (Foster 2002) where the shared resources are computing resources. Trying to create a rich resource pool by aggregating the participants' resources is called resource federation. To this end, and in order to continuously manage the resources, formally defined resource sharing strategies are needed to guide the decisions and actions of individuals or groups. This task is left to the VO administrator who is responsible for the resource allocation. Even in the case of the grid, it is common for resources to be managed on a central approach basis, by local resource management systems such as Condor (Litzkow *et al*. 1988) and PBS (Bode *et al*. 2000).

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

Every participant has different interests and aims (self-interest) when contributing resources to a VO, and much effort and time are spent to generate a mutual acceptance between VO administrator and participants who demand access to a particular resource. There should be strategies to support the cooperative resource federation and to facilitate transparent sharing and load balancing. An automated, policy-compromising method is needed to reduce the workload for the VO administrator in resource federation and to reach satisfied participants (clients).

A VO could enforce policies with static rules, but that would yield sub-optimal performance (e.g., wasteful resource utilization) and therefore, to tackle the above problem, negotiation approaches are popular (Sim 2010). Under this approach, one could enumerate game-theoretical, market-driven methods (Sim 2006), multiple criteria methods (Matsatsinis and Delias 2004; Wanyama and Homayounfar 2007), Open Grid Services Architecture (OGSA)-based methods (Hung *et al*. 2004), combinations of the above or even other intelligent techniques (Cheng *et al*. 2010). A dominant feature in the negotiation paradigm is multiple agents (Faratin *et al*. 1998).

In this work, we claim that the same results could be also achieved in a different way: Indicating candidates to join the VO according to the collectively preferable resources could pointedly support sharing and load balancing. An equivalent case is Infrastructure as a Service (IaaS) cloud computing. In that case the decision that should be made concerns the choice of the *application environment*, which should be hosted in order to have an efficient, and cost effective data center.

Although the proposed solution is not directly comparable with the existing works cited in the previous paragraph, our approach emphasizes:

(1) The policy satisfaction for both parties (VO administrator and participants or data center owner and clients) since participants (clients) stresses self-interested resource utilization while global resource utilization is important for a VO (data center). The key element to address this issue is to derive collective utility functions consistent with individual preferences while reflecting participants' significance. The goal is to improve chances for a satisfied federation (or equivalently for an efficient, and cost effective data center).

(2) Handling the incompleteness of preferences in terms of allowing participants (clients) to express them through an agile framework. Participants declare just some simple preferences relationships while quantitative assessments are derived following a disaggregation–aggregation (D-A) approach.

The rest of the chapter is organized as follows: the next section, is a concise description of the intuition behind using multiple criteria decision aid (MCDA) in agent modeling and multi-agent systems design. Section 11.3 illustrates the problem and its mathematical formulation while some experimental results are presented in Section 11.4. Finally, a concluding section summarizes the chapter and discusses some inherent limitations.

11.2 The intuition of MCDA in multi-agent systems

According to Wooldridge and Jennings (2009) there are three main ways to build agents (three major categories for agents' architectures): the deliberative one, which is the dominant; the reactive; and the median way – the hybrid approaches. In these three broad categories a lot of approaches have been proposed (Bordini 2009). To those

architectures that agents need to be built as some kind of deliberative entities, two main problems appear:

(1) The transduction problem: how to symbolically represent the world, the agents' universe.

(2) The representation/reasoning problem: how to adequately represent information and make useful decisions based on that representation.

Roy's seminal multiple criteria modeling methodology (Roy 1996) can address both these issues. This methodology is ultimately expressed in four levels:

- Level 1: object of the decision, including the definition of the set of potential actions A and the determination of a problematic applied on A.
- Level 2: modeling of a consistent family of criteria assuming that these criteria are non decreasing value functions, exhaustive and nonredundant.
- Level 3: development of a global preference model, to aggregate the marginal preferences on the criteria.
- Level 4: decision-aid or decision support, based on the results of level 3 and the problematic of level 1.

Let us demonstrate in a few lines how this methodology can be applied to the agents modeling problems: a consistent family of criteria (level 2) can sufficiently model the world−with respect to the object of the decision (level 1). A global preference model (level 3) is able to guide agents' decisions. Finally, agents' actions will be realized based upon the decision support step (level 4).

Multiple criteria methodologies could contribute as a methodological background for agents modeling as well as a tool for their real-time implementation (Hutchison *et al*. 2006). However, in this work we stress their potential for modeling support. In this respect, the contribution potentials lie across two levels:

Agent design level: The multiple criteria modeling can respond to the representation issue discussed earlier. Existing methods can provide agents with critical decision making elements.

Multi agent level: Existing methods can profile decision makers (DMs), thus facilitate coordination – negotiation. In addition, DM's profiles can be exploited to assign roles or tasks to agents.

In a previous work (Delias and Matsatsinis 2007), the benefits of using the D-A approach in multi-agent systems design were described. Table 11.1 summarizes these benefits, albeit broadening the perspective to wider MCDA context (not just the D-A paradigm).

11.3 Resource federation applied

11.3.1 Describing the problem in a cloud computing context

In a VO, participants act the part of *clients* while the role of *provider* is left to for the VO administrator. Narrow the available resources into computing resources (e.g., CPU, disk storage, bandwidth, memory), and the VO becomes the equivalent of a data center.

Table 11.1 MCDA contribution potentials to multi-agent systems.

	Issue	Contribution
Agent level	Represent information	Criteria modeling – Criteria must fulfill specific properties
	Make decisions based on the representation	Preference Model Construction
Multi-agent level	Negotiation protocol	Structured process
	Negotiation issues	Criteria trade-offs
	Negotiation reasoning model	Multiple criteria linear function (Preference model)
	Task/role assignment	A global criterion as 'fitness measure'

In that case, the VO 'sells' its infrastructure as a service (IaaS cloud computing). A crucial decision that can have a significant impact on the VO's attractiveness is which *application environments* are hosted by its physical or virtual machines. Clients have the right to demand service level agreements for the application environment that is allocated to them, according to their preferences (which are usually high demanding for some parameters and less demanding for others). On the other hand, the VO administrator (or the data center owner) is expected to aim at the maximization of its profit, as calculated by revenue minus expenses, revenue being contingent on the number of application environments hosted and on the percentage of their faultless performance (according to the quality parameters) and expenses being contingent on equipment cost, energy consumption, financial expenses, etc. Since nearly all expenses are proportional to the number of computational units (Koomey *et al*. 2009), the goal is to minimize their number, i.e., to provide the smallest possible number of application environments that does not violate the clients' service level agreements and guarantees their satisfaction.

Thus the problem emerges as a configuration of the application environments problem. In particular, what CPU, disk storage, bandwidth, memory levels should be assigned to an application environment in order to satisfy all parties? *What are the criteria levels that are collectively the most preferable?* (The equivalent question in the more general case of a VO would be: which participant would be more favored to join the VO based on the resources offered?)

11.3.2 Problem modeling

Let m be the number of existing clients of a data center, namely those who use its infrastructure as a service on a contract basis. These clients act as autonomous, self-interest agents. The notation $\mathbf{C} = \{C_1, C_2, \ldots, C_m\}$ shall be used to symbolize them. Every agent (client) C_t has a weight of significance w_t for the data center owner. This weight could represent the relative value that every agent has for the data center or it could be a parameter declared in the contract. In any case, there should always be $\sum_{t=1}^{m} w_t = 1$.

Let n be the number of criteria $\mathbf{G} = \{g_1, g_2, \ldots, g_n\}$, which will be used to evaluate the alternative solutions. The alternative solutions set can be of any finite size and it shall

be notated as $\mathbf{A} = \{a, b, \ldots\}$. Alternative solutions in this chapter are nothing else than configurations of the application environments, i.e., specific levels of performance on the evaluation criteria. Besides the existing solutions, the methodology suggested in this work introduces a set of reference alternatives $\mathbf{A}_R$. According to Siskos *et al*. (2005) this set could be: a set of past decision alternatives past actions; a subset of decision actions, especially when $\mathbf{A}$ is large; or a set of fictitious actions, consisting of performances on the criteria, which can be easily judged by agents to perform global comparisons.

The concept of reference alternatives is common in the D-A paradigm of the MCDA, however, the novelty of this method is in *not demanding a complete comparisons table*. In particular, every agent (client) is asked to express his/her preferences over just a subset of these reference alternatives. Representing by $\mathbf{A}_{Rt}$ the set of the reference alternatives used for comparisons by the tth agent, the following should hold: $\mathbf{A}_R = \mathbf{A}_{R1} \cup \mathbf{A}_{R2} \cup \ldots \cup \mathbf{A}_{Rm}$. In order to compare alternatives, let us denote an outranking relation S on $\mathbf{A} \times \mathbf{A}$, in a way that $a\ S\ b$ means 'alternative a is at least as good as b'.

The ultimate goal of the methodology is to model the collective preferences of agents (clients). To this end an additive value function u is introduced as follows: $u(\underline{g}) = \sum_{j=1}^{n} u_j(g_j)$. Each $u_j(g_j)$ is piecewise linear on $G_j = \{g_j^1, g_j^2, \ldots, g_j(\alpha_j)\}$, α_j being the number of level of performance of the jth criterion. In addition, the worst and the best performance have standard values as: $u_j(g_{j*}) = 0$ and $\sum_{j=1}^{n} u_j(g_j^{\alpha_j}) = 1$. Finally, the outranking relation is expressed on a value function basis as: $a\ S\ b \Leftrightarrow u[\underline{g(a)}] - u[\underline{g(b)}] \geq 0$.

Figure 11.1 captures how the key participants of the proposed methodology interact with each other. In particular, it illustrates the choreography among the VO administrator and the clients (agents) using the BPMN 2.0 notation. It should be noted that the collapsed activity (Negotation) can iterate until results are satisfying. If this is not possible, then the process terminates with failure.

11.3.3 Assessing agents' value function for resource federation

Each agent provides just two basic pieces of information: the first consists of a set of pairwise comparisons of some reference alternatives. These comparisons are made in terms of the outranking relation defined in the previous section. This way, the t^{th} agent

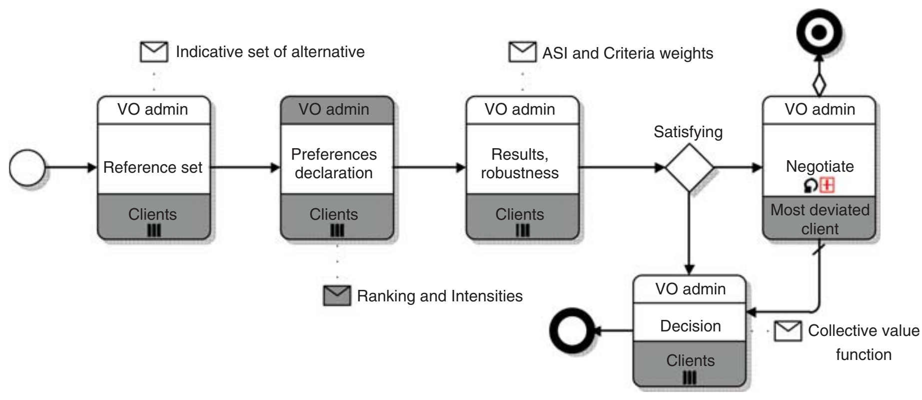

Figure 11.1 Choreography diagram.

provides a *comparisons* set $\mathbf{R}_t \subseteq \mathbf{A}_R \times \mathbf{A}_R$, which could be of any size and include any reference alternatives. A comparison in that set (a row of the matrix) would indicate two alternatives (e.g., a and b) for which the outranking relation $a\ S\ b$ holds. The second piece of information needed is a set of *intensities* about the outranking relations between couples of alternatives of $\mathbf{A}_{Rt}$. Again, this comparisons' set does not have to be complete. More specifically, let $\mathbf{I}_t$ be the set of the '*intensities*' of the tth agent. An element of $\mathbf{I}_t$ would declare if a comparison (an element in $\mathbf{R}_t$) is more 'intense' than any other element in $\mathbf{R}_t$. For example, $a\ S\ b$ is more intense than $c\ S\ d$.

The collective value function will be calculated through a linear regression problem. To this end, two variables z_{tk} and y_{tp} are introduced. The former refers to the kth outranking relationship of the tth agent and the latter to the pth intensity declared by the tth agent. The linear problem is formulated as follows:

$$
\begin{array}{lll}
\min & z = \sum_{t=1}^{m} \left(w_t \sum_{k=1}^{|R_t|} z_{tk} + w_t \sum_{p=1}^{|I_t|} y_{tp} \right) & \\
\text{s.t.:} & u[\underline{g}(a)] - u[\underline{g}(b)] + z_{tk} \geq 0, & t = 1, \ldots, m \\
 & & k = 1, \ldots, |R_t| \\
 & \left(u[\underline{g}(a)] - u[\underline{g}(b)]\right) - \left(u[\underline{g}(c)] - u[\underline{g}(d)]\right) & t = 1, \ldots, m \\
 & \qquad + y_{tp} \geq 0, & \\
 & & p = 1, \ldots, |I_t| \\
 & u_j\left(g_j^{l+1}\right) - u_j\left(g_j^l\right) \geq 0, & j = 1, \ldots, n, \\
 & & l = 1, \ldots, \alpha_j - 1 \\
 & u_j\left(g_j^1\right) = 0 & \\
 & \sum_{j=1}^{n} u_j\left(g_j^{\alpha_j}\right) = 1 & \\
 & u_j\left(g_j^l\right) \geq 0, & j = 1, \ldots, n \\
 & & l = 1, \ldots, \alpha_j \\
 & z_{tk} \geq 0,\ y_{tp} \geq 0, & t = 1, \ldots, m \\
 & & k = 1, \ldots, |R_t| \\
 & & p = 1, \ldots, |I_t|
\end{array}
\tag{11.1}
$$

11.3.3.1 Robustness analysis

Robustness analysis of the results provided by the linear problem is considered as a post-optimality analysis problem. What is actually applied is a slight alteration of the polyhedron defined by the constraints of the initial linear problem. The polyhedron is augmented by the additional constraint $\zeta \leq \zeta^* + \varepsilon,\ \zeta^*$ being the minimal error of the initial linear problem, and ε a very small positive number. A number of $T = \sum_{j=1}^{n}\left(\alpha_j - 1\right)$ new linear problems are constructed and T value functions are calculated by maximizing and minimizing each value $u_j(g_j^l)$, $j = 1, 2, \ldots, n;\ l = 2, \ldots, \alpha_j$, on the augmented polyhedron.

As a measure for the robustness of the marginal value functions the average stability indices are used. An average stability index $ASI(i)$ is the mean value of the normalized

standard deviation of the estimated marginal values on ith criterion and is calculated as

$$ASI(i) = 1 - \frac{1}{\alpha_i - 1} \frac{\sum_{k=1}^{\alpha_i - 1} \sqrt{\left(T \left(\sum_{j=1}^{T} \left(u_k^j\right)^2 \right) - \left(\sum_{j=1}^{T} u_k^j \right)^2 \right)}}{\frac{T}{\alpha_i - 1} \sqrt{(\alpha_i - 2)}} \tag{11.2}$$

where u_k^j is the estimated value of the kth parameter in the jth additive value function.

The global robustness measure will be the average of $ASI(i)$ over all the criteria. If robustness measures are judged satisfactory, i.e., ASI indices are close to 1, then the final solution is calculated as the *barrycentral* value function. Otherwise, the sets $\mathbf{R}_t$ and $\mathbf{I}_t$ should be enriched for one or more agents. The way to guide the $\mathbf{R}_t$ and $\mathbf{I}_t$ redefinition process is by checking the magnitude of the variables z_{tk} and y_{tp}. In particular, the larger these variables are, the greater the inconsistency they will prompt. So, agent t (who is related to z_{tk} and y_{tp}) shall be *contacted by priority*.

11.4 An illustrative example

The proposed methodology is exemplified in this section through the following case: consider a data center that strives to optimize the application environments that it hosts. For the sake of simplicity, let us assume that each application environment is configured around just three criteria: *CPU, disk storage*, and *memory*. In addition, let us assume that the owners of the data center realize the need to negotiate with their clients over some rough levels of performance over those criteria, therefore they use verbal / descriptive scales of criteria measurement. In particular, they use a 3-level scale for the *CPU* and *memory* criteria (e.g., low, average, high) and a 4-level scale for the *disk storage* criterion.

Five agents (clients) are currently negotiating a contract with the data center, i.e., to get hosted in an application environment that best fits their needs. The data center owners have estimated (e.g., according to the contract value) the following significance weights for all the clients: $C_1 = 0.22$, $C_2 = 0.18$, $C_3 = 0.15$, $C_4 = 0.18$ and $C_5 = 0.27$. Recall that the sum of all weights should equal 1. Suppose also, that clients are given a set of indicative alternatives and that they are asked to express their preferences for them. This reference set of alternatives along with their performance over the criteria vector is presented in Table 11.2.

In Table 11.2 the preferences set for each agent contains the outranking relations he/she declares (for instance C_1 has declared that 'alternative a is at least as good as c', 'alternative b is at least as good as c' and 'alternative d is at least as good as e'). The intensities matrix contains as many rows as the number of intensities declared (rows are separated by columns). Each row contains two numerical values, which correspond to the indices of the preferences relations involved. For example, C_1 has declared that the first relation ($a\ S\ c$) is more intense than the second one ($b\ S\ c$) and the third one ($d\ S\ e$). The interesting part is that agents *do not need* to express their preferences over the entire set of alternatives nor do they need to declare intensities for *every pair* of relations. This is an important advantage of the proposed method that provides great flexibility to both the DMs (here the data center owners) and agents.

Table 11.2 Alternatives performance over the criteria vector.

	a	*b*	*c*	*d*	*e*
CPU	3	2	2	2	1
Disk storage	3	3	3	2	4
Memory	2	3	2	2	1

Table 11.3 Agents' preferences.

Agent	Preferences	Intensities
C_1	$\{a\ S\ c,\ b\ S\ c,\ d\ S\ e\}$	[1, 2; 1, 3]
C_2	$\{a\ S\ b,\ b\ S\ c,\ c\ S\ d,\ d\ S\ e\}$	[1, 3; 2, 3; 2, 4]
C_3	$\{b\ S\ a,\ b\ S\ c,\ e\ S\ d\}$	[2, 1]
C_4	$\{a\ S\ d,\ b\ S\ c,\ a\ S\ e,\ d\ S\ e\}$	[2, 3; 2, 4; 3, 4]
C_5	$\{c\ S\ d,\ c\ S\ e,\ b\ S\ e\}$	[1, 2; 3, 1]

Table 11.2 and Table 11.3 contain all the necessary input for the method. If the linear program (11) is applied, the results shown in Table 11.4 are obtained.

The critical results consist of the assessment of the collective preferences (in terms of criteria weights and their marginal utility functions) and of their robustness measures. For instance, the above results indicate that the most important factor for clients is the disk storage capacity of the application environment. In the above example, it is also shown that the final solution is *not robust* enough (the *ASI* index is not close to 1). In this case, data center owners should sum the variables z_{tk} and y_{tp} over t (i.e., for every agent) and ask the one with the maximum value to enrich its preferences statements (i.e., sets $\mathbf{R}_t$ and $\mathbf{I}_t$). In this example case, this agent is C_5. After redefining/adjusting C_5's statements, the linear program shall be resolved and new results shall be calculated. In particular, C_5 declares an additional preference ($a\ S\ e$) in order to make his/her point clearer. Moreover, he/she redefines his/her set of intensities as $\{c\ S\ d,\ c\ S\ e,\ b\ S\ e,\ a\ S\ e\} - [2, 1; 3, 2; 3, 1; 4, 2; 4, 3]$. With this much richer information to hand, the new *ASI* index is 0.612 while new criteria weights are estimated (24.2% for the *CPU*, 43.2% for the *disk storage* and 32.6% for the *memory*). It seems that through this negotiation step, the significance of the *disk storage* criterion is slightly decreased in favor of the other two criteria, in other words agent C_5 agrees to consider the disk storage criterion to be a little less important! If after this step, the *ASI* index is still not satisfactory, the process iterates by contacting the next agent. Finally, and when a

Table 11.4 Example results.

	Weight of significance	Global robustness
CPU	25.7%, $u = [0.0; 0.615; 1]$	
Disk storage	44.8%, $u = [0.0; 0.253; 0.383; 1]$	0.502
Memory	29.5%, $u = [0.0; 0.838; 1]$	

satisfying *ASI* index is reached, data center owners could evaluate different configuration scenarios over the multiple criteria and rank them according to their global performance.

11.5 Conclusions

In this chapter a multicriteria methodology to support resource federation in a cloud computing context was presented. The reasoning components are guided by are the collective preferences of all agents. Therefore, the final solution depends in a very direct way on the agent's rationality. This infuses the system with an impressive flexibility but also with a disagreeable subjectivity. More specifically, modeling agents as rational optimizers based on the suggested multiple criteria approach gives rise to the same limitations as those of classical decision aid: there is a fuzzy borderline between what is and what is not feasible in real decision making contexts; the DMs have seldom well shaped preferences. 'In and among areas of firm convictions lie hazy zones of uncertainty, half held belief, or indeed conflicts and contradictions' (Roy 2005); many data are imprecise, uncertain, or ill-defined. In addition, sometimes, data may not be reflected appropriately into linear utility functions. Furthermore, in a computational context, such as the agents' one, we shall not neglect complexity and time issues: decisions have to be made in real time.

Despite the above limitations, the multiple criteria paradigm emerges as an endeavor to make an objective place for agents' decisions. It provides a way to formalize proactiveness guiding agents to rational and transparent decisions.

References

Bode B, Halstead DM, Kendall R, Lei Z and Jackson D (2000) The portable batch scheduler and the maui scheduler on linux clusters. In *Proceedings of the 4th Annual Linux Showcase and Conference*. USENIX Association, Berkeley, CA, p. 27.

Bordini R (2009) *Multi-agent Programming: Languages, Tools and Applications*. Springer, New York.

Cheng W, Ooi B and Chan H (2010) Resource federation in grid using automated intelligent agent negotiation. *Future Generation Computer Systems* **26**(8), 1116–1126.

Delias P and Matsatsinis NF (2007) The multiple criteria paradigm as a background for agent methodologies. In *8th Annual International Workshop 'Engineering Societies in the Agents World'* (eds Artikis A, O'Hare G, Stathis K and Vouros G), pp. 227–237.

Faratin P, Sierra C and Jennings NR (1998) Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems* **24**(3-4), 159–182.

Foster I (2002) The grid: A new infrastructure for 21st century science. In *Wiley Series in Communications Networking & Distributed Systems* (eds Berman F, Fox G and Hey T). John Wiley & Sons, Ltd, Chichester, pp. 51–63.

Hung P, Haifei L and Jun-Jang J (2004) WS-Negotiation: an overview of research issues. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, p. 10.

Hutchison D, Kanade T, Kittler J, Kleinberg JM, Mattern F, Mitchell JC, Naor M, Nierstrasz O, Rangan CP, Steffen B, Sudan M, Terzopoulos D, Tygar D, Vardi MY, Weikum G, Tučník P, Kožaný J and Srovnal V (2006) Multicriterial Decision-Making in multiagent systems. In *Computational Science – ICCS 2006* (eds Alexandrov VN, Albada GD, Sloot PMA and Dongarra J), vol. 3993. Springer, Berlin, pp. 711–718.

Koomey JG, Belady C, Patterson M, Santos A and Lange KD (2009) Assessing trends over time in performance, costs, and energy use for servers. Microsoft, Intel and Hewlett-Packard Corporation, Technical Report.

Litzkow MJ, Livny M and Mutka MW (1988) Condor-a hunter of idle workstations. *8th International Conference on Distributed Computing Systems*, pp. 104–111.

Matsatsinis N and Delias P (2004) A multi-criteria protocol for multi-agent negotiations. In *Methods and Applications of Artificial Intelligence* (eds Vouros G and Panayiotopoulos T), vol. 3025 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 103–111.

Roy B (1996) *Multicriteria Methodology for Decision Aiding*. Kluwer, Dordrecht.

Roy B (2005) Paradigms and challenges. In *Multiple Criteria Decision Analysis: State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M), vol. 78 of *International Series in Operations Research & Management Science*. Springer, New York, pp. 3–24.

Sim KM (2006) Grid commerce, Market-Driven G-Negotiation, and grid resource management. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **36**(6), 1381–1394.

Sim KM (2010) Grid resource negotiation: Survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **40**(3), 245–257.

Siskos J, Grigoroudis E and Matsatsinis N (2005) UTA methods. In *Multiple Criteria Decision Analysis-State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, Boston, pp. 297–343.

Wanyama T and Homayounfar B (2007) A protocol for multi-agent negotiation in a group-choice decision making process. *Journal of Network and Computer Applications* **30**(3), 1173–1195.

Wooldridge M and Jennings NR (2009) Intelligent agents: theory and practice. *The Knowledge Engineering Review* **10**(02), 115.

12

Fuzzy analytic hierarchy process using type-2 fuzzy sets: An application to warehouse location selection

İrem Uçal Sarı, Başar Öztayşi, and Cengiz Kahraman
Department of Industrial Engineering, Istanbul Technical University, Turkey

12.1 Introduction

In the case of decision making either with a single decision maker or a group of decision makers, it is necessary to take into consideration various points of view. Usually alternatives represent the different choices of action available, and these alternatives should be evaluated based on different viewpoints. A specific criterion can be associated with each point of view, and can be used to assess alternatives on an appropriate qualitative or quantitative scale. Since each criterion represents different points of view, they may conflict with each other, for instance a criterion about investing money may conflict with short-term profit criteria. Another nature of multicriteria decision making (MCDM) is the consideration of incommensurable units in a problem (Triantaphyllou 2000). For instance in a case of buying a house, the cost criterion may be measured in dollars, while the 'building age' criteria is measured in years. Besides the objective measurement scales, the criterion may contain subjective and linguistic evaluations, such as 'security' and 'quality of the neighborhood'. In a decision making problem,

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

using a multicriteria approach is important because it structures the decision problem considering all related aspects, alternatives can be evaluated from various perspectives without any fictitious conversions, and weights, aspiration levels and rejection levels can be determined for each criterion separately (Roy 2005).

In the traditional formulation of MCDM problems human judgments are represented as exact numbers. However, in many practical cases, the data may be imprecise, or the decision makers might be unable to assign exact numerical values to the evaluation. Since some of the evaluation criteria are subjective and qualitative in nature, it is very difficult for the decision maker to express the preferences using exact numerical values (Tseng *et al*. 2008). For instance, the selection of a warehouse location includes both quantitative and qualitative criteria. The qualitative criteria include proximity to support services, quality and reliability of modes of transportation proximity to major highways. The quantitative criteria include transportation costs, construction costs and cost of land and buildings. The conventional MCDM approaches tend to be less effective in dealing with the imprecise or vagueness nature of the linguistic assessment (Kahraman *et al*. 2003).

Fuzzy set theory was specifically designed to mathematically represent uncertainty and vagueness and provide formalized tools for dealing with the imprecision in decision-making problems. Fuzzy sets can appropriately represent imprecise parameters, and can be manipulated through different operations on fuzzy sets. Since imprecise parameters are treated as imprecise values instead of precise ones, the process will be more powerful and its results more credible (Kahraman *et al*. 2006).

Fuzzy set theory is being recognized as an important problem modeling and solution technique because it use approximate information and uncertainty to generate decisions. Knowledge can be expressed in a more natural way by using fuzzy sets, so many decision problems can be greatly simplified (Kahraman and Kaya 2010a). The use of fuzzy set theory as a methodology for modeling and analyzing decision systems is of particular interest to researchers due to the ability of fuzzy set theory to quantitatively and qualitatively model problems which involve vagueness and imprecision (Kahraman *et al*. 2006).

In the classical set theory, the membership of elements in a set is assessed in a binary form: an element either belongs or does not belong to the set. In type-1 fuzzy sets (Zadeh 1965), each element has a degree of membership which is described by a membership function valued in the interval [0, 1]. Although type-1 fuzzy sets can handle uncertainties, some shortcomings are defined in the literature. Mendel and John (2002) define sources of uncertainties that type-1 fuzzy sets cannot handle. These sources are (1) usage of words (words may mean different things to different people), (2) the group of experts that do not all agree, and (3) the noisy data that are analyzed. type-1 fuzzy sets are not capable of modeling such uncertainties since their membership functions are crisp. In order to handle these kinds of uncertainties Zadeh (1975) introduced type-2 fuzzy sets as fuzzy sets that have a fuzzy membership function. Unlike a type-1 fuzzy set where the membership grade is a crisp number, the membership values of type-2 fuzzy sets are also fuzzy numbers themselves. While, the membership functions of type-1 fuzzy sets are two-dimensional, membership functions of type-2 fuzzy sets are three-dimensional. It is the new third dimension that provides additional degrees of freedom that make it possible to directly model uncertainties.

The aim of this chapter is to show the potential applicability of type-2 fuzzy sets in multicriteria selection problems, using warehouse selection as an example. The originality of this chapter comes from the usage of type-2 fuzzy sets on fuzzy AHP with a new

linguistic scale for the first time. In this manner the chapter is organized as follows. In Section 12.2 some multicriteria selection methods are introduced. Section 12.3 describes the fuzzy analytic hierarchy process (AHP) and provides a brief literature review about previous studies. Section 12.4 contains a fuzzy AHP methodology using type-1 fuzzy sets. Type-2 fuzzy sets are introduced in Section 12.5 and the main steps involved in the application of type-2 fuzzy AHP are presented in Section 12.6. Section 12.7 contains a fuzzy AHP application for a warehouse selection problem and finally further steps are discussed in the conclusions.

12.2 Multicriteria selection

MCDM is one of the most well-known branches of decision making, which can be divided into multi-objective decision making (MODM) and multi-attribute decision making (MADM; Zimmermann 1996). However, the terms MADM and MCDM are often used to mean the same class of models. While continuous decision space is available in MODM, MCDM (MADM) concentrates on problems with discrete decision space in which the decision alternatives have been predetermined (Triantaphyllou 2000).

In a decision making process, even when there is a single decision maker, it is rare for him or her to have a single clear criterion. In the case of group decision making, it is nearly impossible for a single well-defined criterion acceptable to the group members to be used in the decision making process. Thus, both in the single or group decision making process it is necessary to take into consideration various points of view, such as capital, human resources, manufacturing, quality, and environmental aspects. A specific criterion can be associated with each point of view, and these criteria can be used to evaluate any potential action or alternative on an appropriate qualitative or quantitative scale. The use of multiple criteria contributes to the decision making process by: (i) structuring the decision process well with all related aspects of the realism with regard to the decision makers involved, (ii) constructing a set of criteria which preserves the original concrete meaning of the corresponding evaluations without having to make any fictitious conversions, (iii) facilitating the debate on the respective role (weight, veto, aspiration level, rejection level) so that each criterion may be discussed by the decision makers during the decision making process (Roy 2005).

MCDM may be considered as a complex and dynamic process including one managerial level and one engineering level (Duckstein and Opricovic 1980). The managerial level defines the goals, and chooses the final ‘optimal’ alternative. The engineering level of the MCDM process defines alternatives and points out the consequences of choosing any one of them from the standpoint of various criteria. This level also performs the multicriteria ranking of alternatives. The main steps of MCDM are the following:

(1) Establishing a system of evaluation criteria that relate system capabilities to goals.

(2) Developing alternative systems for attaining the goals (generating alternatives).

(3) Evaluating alternatives in terms of criteria (the values of the criterion functions).

(4) Applying a normative multicriteria analysis method.

(5) Accepting one alternative as ‘optimal’ (preferred).

(6) If the final solution is not accepted, gather new information and go into the next iteration of multicriteria optimization.

There are various MCDM methods available in the literature which can be classified according to different attributes. According to the type of the data they use, they can be classified as deterministic, stochastic or fuzzy. The methods can be classified according to the number of decision makers involved in the decision process. Chen and Hwang (1991) provide the taxonomy of MCDM methods given in Figure 12.1. Table 12.1 lists some popular MCDM methods along with some recent relevant references.

<table>
<tr><td rowspan="5">Multi-
criteria
decison
making</td><td>Type of information from the decision maker</td><td>Salient feature of information</td><td>Major classes of the methods</td></tr>
<tr><td rowspan="2">No information</td><td></td><td>Dominance
Minimax
Maximax</td></tr>
<tr><td>Standard level</td><td>Conjunctive
Disjunctive</td></tr>
<tr><td rowspan="2">Information on attributes</td><td>Ordinal</td><td>Elimination byaspect
Lexicographic semi order
Lexicographic method</td></tr>
<tr><td>Cardinal</td><td>Weighted sum model
Analytic hierarchy process
ELECTRE
TOPSIS</td></tr>
</table>

Figure 12.1 A taxonomy of MCDM methods.

Table 12.1 MCDM methods and sample references.

Methods	Sample references
ELECTRE	Wu and Chen (2011), Hatami-Marbini and Tavana (2011), Brito *et al*. (2010), Bojković *et al*. (2010), and Montazer *et al*. (2009)
PROMETHEE	Almeida and Vetschera (2012), Vetschera and Almeida (2012), Yilmaz and Dağdeviren (2011), Behzadian *et al*. (2010), and Halouani *et al*. (2009)
TOPSIS	Chen (2012), Büyüközkan and Çifçi (2012), Krohling and Campanharo (2011), Torlak *et al*. (2011), and Celik *et al*. (2009)
WSM	Dong and Hayes (2012), Ramesha *et al*. (2012), Mateo (2012), Zolghadri *et al*. (2011), and Vázqueza *et al*. (2008)
MAUT	Brito and Almeida (2012), Malak *et al*. (2009), Kang and Feng (2009), Løkena *et al*. (2009), and He and Huang (2008)
AHP	Lee *et al*. (2012), Bulut *et al*. (2012), Javanbarg *et al*. (2012), Rajput *et al*. (2011), and Kaya and Kahraman (2011a)

AHP, analytic hierarchy process; MAUT, multi-attribute thoery; WSM, weighted sum model.

12.2.1 The ELECTRE method

ELECTRE, developed by Benayoun and Roy at the late 1960s, is classified as an outranking method in MCDM (Triantaphyllou 2000). In the ELECTRE method concordance and discordance indexes are defined as measurements of satisfaction and dissatisfaction for a decision maker in choosing one alternative over another. These indexes are then used to analyze the outranking relations among the alternatives. In the literature there are several versions of ELECTRE which have different characteristics (Roy 2005).

In an MCDM problem with m alternatives and n criteria, the values assigned to the alternatives are denoted by x_{ij}. The first step is to calculate the normalized alternative values denoted as r_{ij}. Using Equation (12.1) the weighted normalized values are calculated.

$$v_{ij} = r_{ij} w_{ij}, \quad \text{where} \quad i = 1, 2, \ldots, m \quad \text{and} \quad j = 1, 2, \ldots, n \tag{12.1}$$

In the next step the concordance and discordance sets are determined. The concordance set $C(a, b)$ is the collection of attributes where alternative a is better than or equal to alternative b. The complement of $C(a, b)$, the discordance set, contains all criteria for which a is worse than b. Depending on these sets concordance and discordance indexes are calculated for each pair of alternatives. And finally outranking relationships are defined to measure the outranking degree for a over b (Yoon and Hwang 1995).

12.2.2 PROMETHEE

PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) belongs to the outranking MCDM methods family. The decision maker forms the preference functions of PROMETHEE by pair wise comparison of the alternatives. For a given criterion j, and considering two alternatives a and b, the difference between them will be equal to the difference of their scores, i.e.:

$$d_j(a, b) = v_j(a) - v_j(b) \tag{12.2}$$

Brans and Mareschal (2005) define six types of preference functions. Entering the value of each alternative one of these functions, a value is found which is between 0 and 1 (Wang *et al*. 2010).

In the flow of PROMETHEE, the decision maker chooses the preference function and the threshold value for each criterion depending on its nature. Then, the decision matrix is formed containing criteria and alternatives. The alternative values are entered in the functions and the value found by the preference function is multiplied by the weight of the criterion. The procedure repeats for all pairs of alternatives, and a pairwise comparison matrix of the alternatives is formed. On the basis of this pairwise comparison matrix the positive and negative flows of each alternative are computed and used to select the best alternative. There are several versions of PROMETHEE in the literature, developed for specific purposes (Roy 2005).

12.2.3 TOPSIS

In TOPSIS (Technique for Order Preference by Similarity to Ideal Situation) the alternatives are evaluated according to their distances from the optimal solution. The alternative

with the shortest distance is selected as the best alternative. In the initial step, the positive and negative ideal solutions are determined. The positive ideal solution (A^+) is determined by selecting the largest normalized and weighted score for each criterion. The negative ideal solution (A^-) is determined by selecting the lowest normalized and weighted score of each criterion (Roy 2005).

In the second step, the distances to the positive ideal solution $(R)^+$ and negative ideal solution $(R)^-$ for each alternative are calculated. These distances are used to compute the closeness index (CI) to the ideal solution:

$$CI = \frac{(R)^-}{(R)^+ + (R)^-} \tag{12.3}$$

The CI varies between 0 and 1 and the alternative with the highest CI value is selected as the best alternative.

12.2.4 The weighted sum model method

The WSM is one of the most commonly used approaches in MCDM. The rationale behind the method relies on additive utility assumption, which is the total value of each alternative is equal to the sum of products:

$$a^* = \max_{\mathrm{i}} \sum_{j=1}^{n} x_{ij} w_j, \quad \text{for } i = 1, 2, \ldots, m \tag{12.4}$$

where a^* is the best alternative, n is the number of criteria, x_{ij} is the actual value of the ith alternative in the terms of the jth criterion, w_j is the weight of criterion j, and m is the number of alternatives. In the case where all the criteria are of the benefit type, the alternative with the highest WSM score is selected as the best alternative (Fishburn 1967). The weakness of the method can be realized when different units such as money, age, millage have to be used in the same decision problem (Triantaphyllou 2000).

12.2.5 Multi-attribute utility theory

In MAUT the concept of utility is user to measure the satisfaction experienced by a person who receives a good or a service. MAUT takes into account uncertainty as well as the risk attitude of the decision maker (Vincke 1992). The decision maker can define utility functions with different shapes depending on his/her attitude towards risk. The functions can be, risk natural, risk-averse or risk-seeking (Wang *et al*. 2010). After assessing the utility function for each criterion, the overall utility of each alternative can be evaluated by the weighted sum of all attribute values of the alternative:

$$U(a_i) = \sum_{j=1}^{n} w_j u_j(x_{ij}) \tag{12.5}$$

where $U(a_i)$ is the utility of alternative i and $u_j(x_{ij})$ denotes the utility gained from alternative i on criterion j. The alternative with the largest integrated utility value is selected as the best alternative.

Table 12.2 Verbal judgments and numerical rate.

Verbal judgment of preference	Numerical rate
Equal importance	1
Weak importance of one over another	3
Essential or strong importance	5
Demonstrated importance	7
Absolute importance	9
Intermediate values between the two adjacent judgments	2, 4, 6, 8

12.2.6 Analytic hierarchy process

The AHP, developed by Saaty (1980), structures a decision problem as a hierarchy, containing an overall goal, a group of alternatives, and of a group of criteria which link the alternatives to the goal.

AHP is based on the use of pairwise comparisons, which lead to the elaboration of a ratio scale. Pairwise comparisons are classically carried out by asking how more valuable an alternative a is to criterion j than another alternative b. The judgments of the decision maker are transformed into numerical values using the defined scales used in AHP. A pairwise comparison matrix is formed using the pairwise comparisons of alternatives that vary between 1/9 and 9. The diagonal elements of the matrix are equal to 1 while the other ones change between the values defined in Table 12.2 and the inverse of these values. The pairwise comparison matrix is processed mathematically, in order to transform user information, into mathematically defined priorities.

Saaty (1990) proposes a consistency index in order to measure the subjective measurements of the decision maker. The consistency index is compared with the same index obtained as an average over a large number of reciprocal matrices of the same order whose entries are random. If the ratio (called the consistency ratio) of the consistency index to that from random matrices is significantly small (carefully specified to be about 10% or less), the priorities are accepted, otherwise the decision maker is asked to revise the pairwise comparisons.

In order to solve the MCDM using AHP the following steps can be followed (Saaty 2008):

(1) A pairwise comparison matrix is formed for comparing the criteria.

(2) The decision maker is asked to make comparisons between the criteria using preference. As the pairwise comparison matrix is generated by the decision maker, the resulting weights are calculated by using the eigenvector approach.

(3) The consistency of the judgments is checked and the comparisons are reviewed if needed.

(4) To weight alternatives, a similar procedure (from 1 to 3) applies. The pairwise comparison matrix of alternatives is formed with respect to one criterion.

The result is a new reciprocal square matrix for each criterion, with its corresponding eigenvector. The procedure is repeated for all criteria and the value of each alternative and criterion is calculated.

(5) The value of each alternative is multiplied by the weight of the corresponding criterion.

(6) Finally, all the values for an alternative are added up, the final calculation results indicate the importance of each alternative.

In the classical AHP approach, human judgments are represented as exact numbers. However, in many practical cases the human preference model is uncertain and decision makers might be reluctant or unable to assign exact numerical values to the comparison judgments. The decision makers are usually unsure in their levels of preference due to incomplete and uncertain information about possible alternatives and their performances. Since some of the evaluation criteria are subjective and qualitative, it is very difficult for the decision maker to express the strength of his/her preferences and to provide exact pairwise comparison judgments. Hence, we use the fuzzy AHP developed by Buckley (1985).

12.3 Literature review of fuzzy AHP

There have been lots of fuzzy AHP methods proposed by various authors in the literature. The first study on fuzzy AHP was presented by van Laarhoven and Pedrycz (1983) where fuzzy scores were defined by triangular membership functions. Buckley (1985) determined fuzzy scores by trapezoidal membership functions and criticized van Laarhoven and Pedryzs' method. Boender *et al*. (1989) presented a more robust approach to the normalization of local priorities. Stam *et al*. (1996) used an artificial neural network in order to determine the preference ratings in AHP. Chang (1996) brought forward the extent analysis method based on the utilization of triangular fuzzy numbers for pairwise comparisons. Cheng (1996) proposed a new algorithm for the assessment of a tactical missile system. Kahraman *et al*. (1998) proposed a fuzzy objective and subjective method based on fuzzy AHP. Cheng *et al*. (1999) presented a new approach for evaluating weapon systems based on linguistic variables. Zhu *et al*. (1999) discussed the extent analysis method and applications of fuzzy AHP. Zeng *et al*. (2007) developed a modified fuzzy AHP for project risk assessment. Leung and Cao (2000) proposed a consistency definition that considers tolerance deviation for the alternatives in the fuzzy AHP. Chan *et al*. (2000) reported a technology algorithm to quantify both tangible and intangible benefits in a fuzzy environment. Tsaur *et al*. (2002) implemented fuzzy AHP in order to assess airline service quality. Rong *et al*. (2003) combined AHP and the fuzzy set theory for enterprise waste evaluation. Bozdag *et al*. (2003) selected the best computer-integrated manufacturing system by four different fuzzy multi-attribute decision making methods. Kahraman *et al*. (2004) implemented the extent analysis method in order to select the best catering firm. Kulak and Kahraman (2005) used fuzzy axiomatic design and fuzzy AHP to make a selection among transportation companies. Haq and Kannan (2006) implemented fuzzy AHP for evaluating suppliers in a supply chain model. Bozbura *et al*. (2007) developed a model to improve the quality of the prioritization of human capital measurement indicators under fuzziness. Kayakutlu and Büyüközkan (2008) used an integrated Delphi – fuzzy AHP based model to prioritize the balancing factors. Li and

Huang (2009) applied TRIZ and fuzzy AHP to develop innovative design for automated manufacturing systems. Gungor *et al*. (2009) proposed a personnel selection algorithm based on fuzzy AHP. Kahraman and Kaya (2010b) used a fuzzy AHP methodology in order to select the best energy alternative. Lin (2010) utilized fuzzy AHP to examine the similarities and differences between high and low experience groups for the evaluation of course website quality. Kutlu and Ekmekçioğlu (2012) proposed a new approach for fuzzy failure mode and effect analysis (FMEA) by applying fuzzy TOPSIS integrated with Chang's extent analysis method for fuzzy AHP. Kaya and Kahraman (2011b) suggested an environmental impact assessment approach in the context of urban planning. The approach is based on an integrated fuzzy AHP–ELECTRE method.

Table 12.3 indicates the differences among the fuzzy AHP methods in the literature, including the advantages and disadvantages, and main characteristic of each method.

In the following section we briefly give the steps of Buckley's type-1 fuzzy AHP.

12.4 Buckley's type-1 fuzzy AHP

AHP is a structured approach to decision making developed by Saaty (1980). It is a weighted factor-scoring model and has the ability to detect and incorporate inconsistencies inherent in the decision making process. Therefore, it has been applied to a wide variety of decision making problems, including the evaluation of alternatives. Sometimes a decision maker's judgments are not crisp, and it is relatively difficult for the decision maker to provide exact numerical values. Therefore most of the evaluation parameters cannot be given as precisely and the evaluation data of the alternative project's suitability for various subjective criteria and the weights of the criteria are usually expressed in linguistic terms by the decision makers. In this case, fuzzy logic that provides a mathematical strength to capture the uncertainties associated with human cognitive process can be used (Birgn *et al*. 2010; Kahraman *et al*. 2010). In this chapter, Buckley's fuzzy AHP method is presented in detail. The steps of this method are:

- Step 1. Pairwise comparison matrices are constructed. Each element ($\tilde{a}_{ij}$) of the pairwise comparison matrix $\tilde{A}_k$ is a linguistic term representing which is the more important of two criteria. The pairwise comparison matrix is given by:

$$\tilde{A}_k = \begin{vmatrix} 1 & \tilde{a}_{12} & \dots & \tilde{a}_{1n} \\ \tilde{a}_{21} & 1 & \dots & \tilde{a}_{2n} \\ \dots & \dots & \dots & \dots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \dots & 1 \end{vmatrix}, \quad k = 1, 2, \dots, K \qquad (12.6)$$

  where $\tilde{A}_k$ is a pairwise comparison matrix for the kth expert for FR_m. For the evaluation procedure, the linguistic terms given in Table 12.4 are used. The geometric mean is used to aggregate expert opinions.

- Step 2. Examine the consistency of the fuzzy pairwise comparison matrices. Assume $A = [a_{ij}]$ is a positive reciprocal matrix and $\tilde{A} = [\tilde{a}_{ij}]$ is a fuzzy positive reciprocal matrix. If the result of the comparisons of $A = [a_{ij}]$ is consistent, then it can imply that the result of the comparisons of $\tilde{A} = [\tilde{a}_{ij}]$ is also

Table 12.3 Comparison of different fuzzy AHP methods (Büyüközkan *et al*. 2004).

Sources	The main characteristics of the method	Advantages and disadvantages
van Laarhoven and Pedrycz (1983)	Direct extension of Saaty's AHP method with triangular fuzzy numbers Lootsma's logarithmic least square method is used to derive fuzzy weight and fuzzy performance scores	(A) The options of multiple experts can be modeled in the reciprocal matrix (D) There is not always a solution to linear equations (D) The computational requirement is high, even for a small problem (D) It allows only triangular fuzzy numbers to be used
Buckley (1985)	Extension of Saaty's AHP method with trapezoidal fuzzy numbers Uses the geometric mean method to derive fuzzy weights and performance scores	(A) It is easy to extend to the fuzzy case (A) It guarantees a unique solution to the reciprocal comparison matrix (D) The computational requirement is high
Boender *et al*. (1989)	Modifies the method of van Laarhoven and Pedrycz Present a more robust approach to the normalization of the local priorities	(A) The options of multiple experts can be modeled (D) The computational requirement is high
Chang (1996)	Synthetical degree values Layer simple sequencing Composite total sequencing	(A) The computational requirement is relatively low (A) It follows the steps of crisp AHP. It does not involve additional operations (D) It allows only triangular fuzzy numbers to be used
Cheng (1996)	Builds fuzzy standards Represents performance scores by membership functions	(A) The computational requirement is high (D) Entropy is used when probability distribution is known (D) The method is based on both probability and possibility measures
Zeng *et al*. (2007)	Uses arithmetic averaging method to get performance scores Extension of Saaty's AHP method with different scales contains triangular, trapezoidal, and crisp numbers	(A) It follows the steps of crisp AHP (A) The options of multiple experts can be modeled (A) There is a flexibility of using different scales (D) The computational requirement is high when there are many expert judgments

(A), advantage; (D), disadvantage.

Table 12.4 Linguistic scale for weight matrix. Reproduced by permission of Taylor & Francis Ltd.

Linguistic scales	Scale of fuzzy number
(1,1,3)	Equally important
(1,3,5)	Weakly important
(3,5,7)	Essentially important
(5,7,9)	Very strongly important
(7,9,9)	Absolutely important

consistent (Buckley, 1985). In order to check the consistency of the fuzzy pairwise comparison matrices, pairwise comparisons are defuzzified by the graded mean integration approach (Chen and Hsieh, 1999). According to the graded mean integration approach, a triangular fuzzy number $\tilde{A} = (l, m, u)$ can be transformed into a crisp number by employing:

$$A = \frac{l + 4m + u}{6}. \tag{12.7}$$

- Step 3. Compute the fuzzy geometric mean for each criterion. Let $(\tilde{r}_i)$ be the geometric mean of each row of $\tilde{A} = [\tilde{a}_{ij}]$ then it is calculated as:

$$\tilde{r}_i = [\tilde{a}_{i1} \otimes \cdots \otimes \tilde{a}_{in}]^{1/n}. \tag{12.8}$$

- Step 4. Compute the fuzzy weights by normalization. If $(\tilde{w}_i)$ is the fuzzy weight of the ith criterion, which is represented by a triangular fuzzy number, $\tilde{w}_i = (lw_i, mw_i, uw_i)$. Here, l and u are the lower and upper bound of the fuzzy weight $\tilde{w}_i$, m is the modal value of $\tilde{w}_i$. The fuzzy weight of the ith criterion is calculated as:

$$\tilde{w}_i = \tilde{r}_i \otimes [\tilde{r}_1 \oplus \cdots \oplus \tilde{r}_i \oplus \cdots \oplus \tilde{r}_n]^{-1}. \tag{12.9}$$

- Step 5. Defuzzification of fuzzy numbers in order to determine the importance ranking of the criteria. In previous works, the procedure of defuzzification has been to locate the best non-fuzzy performance (BNP) value (Hsieh *et al*. 2004). The center of area (COA or center index) method can be used for defuzzification in this step.

The COA method's BNP value for triangular fuzzy number $\tilde{A} = (l, m, u)$ can be calculated as:

$$BNP_i = \frac{(u_i - l_i) + (m_i - l_i)}{3} + l_i, \quad \forall\, i. \tag{12.10}$$

12.5 Type-2 fuzzy sets

Type-2 fuzzy sets are used for modeling uncertainty and imprecision in a better way. Type-2 fuzzy sets are the fuzzy sets which have fuzzy membership functions. Zadeh (1975) introduce type-2 fuzzy sets. Type-2 fuzzy sets are motivated by the close association which exists between the concept of a linguistic truth with truth-values such as true, quite true, very true, more or less true, etc., on the one hand, and fuzzy sets in which the grades of membership are specified in linguistic terms such as low, medium, high, very low, not low and not high, etc., on the other (Zadeh 1975).

In this section, some definitions are presented for type-2 fuzzy sets and interval type-2 fuzzy sets following Mendel *et al*. (2006).

A type-2 fuzzy set $\tilde{A}$ in the universe of discourse X can be represented by a type-2 membership function $\mu_{\tilde{A}}$ shown as follows (Zadeh 1975):

$$\tilde{\tilde{A}} = \left\{ \left((x, u), \mu_{\tilde{\tilde{A}}}(x, u) \right) | \forall x \in X, \forall u \in J_x \subseteq [0, 1], 0 \leq \mu_{\tilde{\tilde{A}}}(x, u) \leq 1 \right\} \tag{12.11}$$

where J_x denotes an interval [0,1]. Moreover, the type-2 fuzzy set $\tilde{\tilde{A}}$ also can be represented as follows (Mendel *et al*. 2006):

$$\tilde{\tilde{A}} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{\tilde{A}}}(x, u) / (x, u), \tag{12.12}$$

where $J_x \subseteq [0, 1]$ and $\iint$ denotes union over all admissible x and u.

Let $\tilde{\tilde{A}}$ be a type-2 fuzzy set in the universe of discourse X represented by the type-2 membership function $\mu_{\tilde{\tilde{A}}}$. If all $\mu_{\tilde{\tilde{A}}}(x, u) = 1$, then $\tilde{\tilde{A}}$ is called an interval type-2 fuzzy set (Zadeh 1975). An interval type-2 fuzzy set $\tilde{\tilde{A}}$ can be regarded as a special case of a type-2 fuzzy set, represented as (Mendel *et al*. 2006):

$$\tilde{\tilde{A}} = \int_{x \in X} \int_{u \in J_x} 1/(x, u), \tag{12.13}$$

where $J_x \subseteq [0, 1]$.

Definition 12.5.1 *The upper membership function and the lower membership function of an interval type-2 fuzzy set are type-1 membership functions, respectively.*

A trapezoidal interval type-2 fuzzy set is illustrated as:

$$\tilde{\tilde{A}}_i = (\tilde{A}_i^U, \tilde{A}_i^L) = \Big((a_{i1}^U, a_{i2}^U, a_{i3}^U, a_{i4}^U; H_1\left(\tilde{A}_i^U\right), H_2\left(\tilde{A}_i^U\right)\Big),$$
$$\Big(a_{i1}^L, a_{i2}^L, a_{i3}^L, a_{i4}^L; H_1(\tilde{A}_i^L), H_2(\tilde{A}_i^L))\Big)$$

where $\tilde{A}_i^U$ and $\tilde{A}_i^L$ $(1 \leq i \leq n)$ are type-1 fuzzy sets, $a_{i1}^U, \ldots, a_{i4}^U, a_{i1}^L, \ldots, a_{i4}^L$ are the reference points of the interval type-2 fuzzy set $\tilde{A}_i$, $H_j(\tilde{A}_i^U) \in [0, 1]$ denotes the membership value of the element $a_{j(j+1)}^U$ $(j = 1, 2)$ in the upper trapezoidal membership function $\tilde{A}_i^U$, and $H_j(\tilde{A}_i^L) \in [0, 1]$ denotes the membership value of the element $a_{j(j+1)}^L$ in the lower trapezoidal membership function $\tilde{A}_i^L$ (Chen and Lee 2010).

Definition 12.5.2 *The addition operation between two trapezoidal interval type-2 fuzzy sets $\tilde{\tilde{A}}_1$ and $\tilde{\tilde{A}}_2$ is defined as follows (Chen and Lee* 2010*):*

$$\begin{aligned}\tilde{\tilde{A}}_1 \oplus \tilde{\tilde{A}}_2 &= \left(\tilde{A}_1^U, \tilde{A}_1^L\right) \oplus \left(\tilde{A}_2^U, \tilde{A}_2^L\right)\\ &= \big(\big(a_{11}^U + a_{21}^U, a_{12}^U + a_{22}^U, a_{13}^U + a_{23}^U, a_{14}^U + a_{24}^U;\\ &\quad \min\left(H_1(\tilde{A}_1^U), H_1(\tilde{A}_2^U)\right), \min\left(H_2(\tilde{A}_1^U), H_2(\tilde{A}_2^U)\right)\big),\\ &= \big(a_{11}^L + a_{21}^L, a_{12}^L + a_{22}^L, a_{13}^L + a_{23}^L, a_{14}^L + a_{24}^L;\\ &\quad \min\left(H_1(\tilde{A}_1^L), H_1(\tilde{A}_2^L)\right), \min\left(H_2(\tilde{A}_1^L), H_2(\tilde{A}_2^L)\right)\big)\end{aligned} \tag{12.14}$$

Definition 12.5.3 *The subtraction operation between two trapezoidal interval type-2 fuzzy sets $\tilde{\tilde{A}}_1$ and $\tilde{\tilde{A}}_2$ is defined as follows (Chen and Lee,* 2010*):*

$$\begin{aligned}\tilde{\tilde{A}}_1 \ominus \tilde{\tilde{A}}_2 &= \left(\tilde{A}_1^U, \tilde{A}_1^L\right) \ominus \left(\tilde{A}_2^U, \tilde{A}_2^L\right)\\ &= \big(\big(a_{11}^U - a_{24}^U, a_{12}^U - a_{23}^U, a_{13}^U - a_{22}^U, a_{14}^U - a_{21}^U\\ &\quad \min\left(H_1(\tilde{A}_1^U), H_1(\tilde{A}_2^U)\right), \min\left(H_2(\tilde{A}_1^U), H_2(\tilde{A}_2^U)\right)\big),\\ &= \big(a_{11}^L - a_{24}^L, a_{12}^L - a_{23}^L, a_{13}^L - a_{22}^L, a_{14}^L - a_{21}^L;\\ &\quad \min\left(H_1(\tilde{A}_1^L), H_1(\tilde{A}_2^L)\right), \min\left(H_2(\tilde{A}_1^L), H_2(\tilde{A}_2^L)\right)\big)\end{aligned} \tag{12.15}$$

Definition 12.5.4 *The multiplication operation between two trapezoidal interval type-2 fuzzy sets $\tilde{\tilde{A}}_1$ and $\tilde{\tilde{A}}_2$ is defined as follows (Chen and Lee,* 2010*):*

$$\begin{aligned}\tilde{\tilde{A}}_1 \otimes \tilde{\tilde{A}}_2 &= \left(\tilde{A}_1^U, \tilde{A}_1^L\right) \otimes \left(\tilde{A}_2^U, \tilde{A}_2^L\right)\\ &= \big(\big(a_{11}^U \times a_{21}^U, a_{12}^U \times a_{22}^U, a_{13}^U \times a_{23}^U, a_{14}^U \times a_{24}^U;\\ &\quad \min\left(H_1(\tilde{A}_1^U), H_1(\tilde{A}_2^U)\right), \min\left(H_2(\tilde{A}_1^U), H_2(\tilde{A}_2^U)\right)\big),\\ &= \big(a_{11}^L \times a_{21}^L, a_{12}^L \times a_{22}^L, a_{13}^L \times a_{23}^L, a_{14}^L \times a_{24}^L;\\ &\quad \min\left(H_1(\tilde{A}_1^L), H_1(\tilde{A}_2^L)\right), \min\left(H_2(\tilde{A}_1^L), H_2(\tilde{A}_2^L)\right)\big)\end{aligned} \tag{12.16}$$

Definition 12.5.5 *The arithmetic operations between a trapezoidal interval type-2 fuzzy set $\tilde{\tilde{A}}_1$ and a crisp value $k > 0$ is defined as follows (Chen and Lee,* 2010*):*

$$\begin{aligned}k\tilde{\tilde{A}}_1 = \big(&(k \times a_{11}^U, k \times a_{12}^U, k \times a_{13}^U, k \times a_{14}^U; H_1(\tilde{A}_1^U), H_2(\tilde{A}_1^U)\big),\\ &\big(k \times a_{11}^L, k \times a_{12}^L, k \times a_{13}^L, k \times a_{14}^L; H_1(\tilde{A}_1^L), H_2(\tilde{A}_1^L)\ )\big)\end{aligned} \tag{12.17}$$

Table 12.5 Definition and the scale of the linguistic variables used.

Linguistic variables	Type-2 fuzzy scales
Absolutely strong (AS)	(7,8,9,9;1,1) (7.2,8.2,8.8,9;0.8,0.8)
Very strong (VS)	(5,6,8,9;1,1) (5.2,6.2,7.8,8.8;0.8,0.8)
Fairly strong (FS)	(3,4,6,7;1,1) (3.2,4.2,5.8,6.8;0.8,0.8)
Slightly strong (SS)	(1,2,4,5;1,1) (1.2,2.2,3.8,4.8;0.8,0.8)
Exactly equal (E)	(1,1,1,1;1,1) (1,1,1,1;1,1)
If factor i has one of the above linguistic variables assigned to it when compared with factor j, then j has the reciprocal value when compared with i	Reciprocals of the above

$$\begin{aligned}\frac{\tilde{\tilde{A}}_1}{k} =& \left(\left(\frac{1}{k} \times a_{11}^U, \frac{1}{k} \times a_{12}^U, \frac{1}{k} \times a_{13}^U, \frac{1}{k} \times a_{14}^U; H_1(\tilde{A}_1^U), H_2(\tilde{A}_1^U)\right),\right. \\ & \left.\left(\frac{1}{k} \times a_{11}^L, \frac{1}{k} \times a_{12}^L, \frac{1}{k} \times a_{13}^L, \frac{1}{k} \times a_{14}^L; H_1(\tilde{A}_1^L), H_2(\tilde{A}_1^L)\right)\right)\end{aligned} \tag{12.18}$$

12.6 Type-2 fuzzy AHP

The linguistic variables and their type-2 fuzzy scales which are used in fuzzy AHP are determined and given in Table 12.5.

The fuzzy AHP procedure for determining the evaluation weights is explained as follows:

- Step 1. Construct fuzzy pairwise comparison matrices among all the criteria in the dimensions of the hierarchy system. The result of the comparisons is constructed as fuzzy pairwise comparison matrices $(\tilde{A})$ as follows:

$$\tilde{\tilde{A}} = \begin{bmatrix} 1 & \tilde{\tilde{a}}_{12} & \cdots & \tilde{\tilde{a}}_{1n} \\ \tilde{\tilde{a}}_{21} & 1 & \cdots & \tilde{\tilde{a}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\tilde{a}}_{n1} & \tilde{\tilde{a}}_{n2} & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 & \tilde{\tilde{a}}_{12} & \cdots & \tilde{\tilde{a}}_{1n} \\ 1/\tilde{\tilde{a}}_{12} & 1 & \cdots & \tilde{\tilde{a}}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/\tilde{\tilde{a}}_{1n} & 1/\tilde{\tilde{a}}_{2n} & \cdots & 1 \end{bmatrix} \tag{12.19}$$

where

$$1/\tilde{a} = \left(\left(\frac{1}{a_4^U}, \frac{1}{a_3^U}, \frac{1}{a_2^U}, \frac{1}{a_1^U}; H_1^U(\tilde{a}), H_2^U(\tilde{a})\right), \left(\frac{1}{a_4^L}, \frac{1}{a_3^L}, \frac{1}{a_2^L}, \frac{1}{a_1^L}; H_1^L(\tilde{a}), H_2^L(\tilde{a})\right)\right).$$

- Step 2. Examine the consistency of the fuzzy pairwise comparison matrices. Assume$A = [a_{ij}]$ is a positive reciprocal matrix and $\tilde{\tilde{A}} = [\tilde{\tilde{a}}_{ij}]$ is a fuzzy positive reciprocal matrix. If the result of the comparisons of$A = [a_{ij}]$ is consistent, then it can imply that the result of the comparisons of$\tilde{\tilde{A}} = [\tilde{\tilde{a}}_{ij}]$ is also consistent.

- Step 3. Compute the fuzzy geometric mean for each criterion. Let $\tilde{\tilde{r}}_i$ be the geometric mean of each row of $\tilde{\tilde{A}} = [\tilde{\tilde{a}}_{ij}]$then it is calculated as:

$$\tilde{\tilde{r}}_i = \left[\tilde{A}_{i1} \otimes \cdots \otimes \tilde{A}_{in}\right]^{1/n} \tag{12.20}$$

where

$$\sqrt[n]{\tilde{A}_{ij}} = \left(\left(\sqrt[n]{a_{ij1}^U}, \sqrt[n]{a_{ij2}^U}, \sqrt[n]{a_{ij3}^U}, \sqrt[n]{a_{ij4}^U}; H_1^u(\tilde{a}_{ij}), H_2^u(\tilde{a}_{ij})\right), \left(\sqrt[n]{a_{ij1}^L}, \sqrt[n]{a_{ij2}^L}, \sqrt[n]{a_{ij3}^L}, \sqrt[n]{a_{ij4}^L}; H_1^L(\tilde{a}_{ij}), H_2^L(\tilde{a}_{ij})\right)\right).$$

- Step 4. Compute the fuzzy weights by normalization. The fuzzy weight $\tilde{\tilde{w}}_i$ of the ith criterion is calculated as:

$$\tilde{\tilde{w}}_i = \tilde{\tilde{r}}_i \otimes \left[\tilde{\tilde{r}}_1 \oplus \cdots \oplus \tilde{\tilde{r}}_i \oplus \cdots \oplus \tilde{\tilde{r}}_n\right]^{-1} \tag{12.21}$$

where

$$\frac{\tilde{\tilde{a}}}{\tilde{\tilde{b}}} = \left(\left(\frac{a_1^u}{b_4^u}, \frac{a_2^u}{b_3^u}, \frac{a_3^u}{b_2^u}, \frac{a_4^u}{b_1^u}, \min\left(H_1^u(\tilde{a}), H_1^u(\tilde{b})\right), \min\left(H_2^u(\tilde{a}), H_2^u(\tilde{b})\right)\right), \left(\frac{a_1^L}{b_4^L}, \frac{a_2^L}{b_3^L}, \frac{a_3^L}{b_2^L}, \frac{a_4^L}{b_1^L}, \min\left(H_1^L(\tilde{a}), H_1^L(\tilde{b})\right), \min\left(H_2^L(\tilde{a}), H_2^L(\tilde{b})\right)\right)\right).$$

- Step 5. Defuzzify fuzzy numbers in order to determine the importance ranking of the criteria. In this step we use the centroid method to defuzzify the upper and lower membership values of type-2 fuzzy numbers, then we calculate the arithmetic average of the results, and rank the type-2 fuzzy numbers according to these results.

12.7 An application: Warehouse location selection

A chemicals manufacturer in Turkey is planning to locate a warehouse in one of three defined locations in the Marmara Region. Locations are defined due to the industrial zones that they have. The alternatives are decided as Bursa, Tekirdağ, and Balikesir. Four factors are determined to select the best location for the warehouse, including security (F1), general location (F2), infrastructure (F3), and cost (F4). Proximity to support services (C1) and potential of having natural disasters (C2) are defined as the criteria of the security

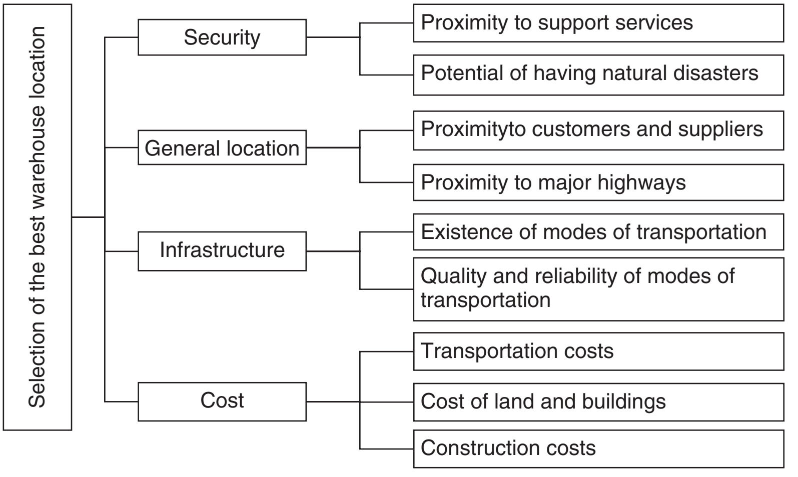

Figure 12.2 Hierarchical structure of factors and criteria.

factor. Proximity to customers and suppliers (C3) and proximity to major highways (C4) are criteria of the general location factor. The infrastructure factor has two criteria which are existence of modes of transportation (C5) and quality and reliability of modes of transportation (C6) and the cost factor has three criteria which are transportation costs (C7), cost of land and buildings (C8), and construction costs (C9). The hierarchical structure of factors and criteria is given in Figure 12.2.

The experts make their evaluations by using the linguistic terms defined in Table 12.4. Then, these linguistic terms are converted to their numerical counterparts. Table 12.6 presents the linguistic variables of the pairwise comparison matrix which are defined by five different experts for the factors.

The elements of the pairwise comparison matrices are calculated by using the geometric mean method as follows:

$$\tilde{\tilde{a}}_{ij} = (\tilde{\tilde{a}}_{ij}^1 \otimes \tilde{\tilde{a}}_{ij}^2 \otimes \tilde{\tilde{a}}_{ij}^3 \otimes \tilde{\tilde{a}}_{ij}^4 \otimes \tilde{\tilde{a}}_{ij}^5)^{1/5}.$$

Table 12.6 Linguistic variables of the pairwise comparison matrix for the factors.

	F1	F2	F3	F4
F1	E, E, E, E, E	FW, FW, SW, FW, FW	FW, SW, FW, FW, SW	VW, VW, VW, FW, AW
F2	FS, FS, SS, FS, FS	E, E, E, E, E	SS, SW, SS, SS, SS	SW, FW, SW, SW, VW
F3	FS, SS, FS, FS, SS	SS, SS, SW, SW, SW	E, E, E, E, E	FW, SW, FW, SW, VW
F4	VS, VS, VS, FS, AS	SS, FS, SS, SS, VS	FS, SS, FS, SS, VS	E, E, E, E, E

For example, the synthetic value for the $\tilde{\tilde{a}}_{12^{\text{th}}}$ element of the pairwise comparison matrix for the dimensions is calculated as:

$$\tilde{\tilde{a}}_{12} = \left(\tilde{\tilde{a}}_{12}^{1} \otimes \tilde{\tilde{a}}_{12}^{2} \otimes \tilde{\tilde{a}}_{12}^{3} \otimes \tilde{\tilde{a}}_{12}^{4} \otimes \tilde{\tilde{a}}_{12}^{5}\right)$$

$$= \left(\left(\frac{1}{7}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}; 1, 1\right)\left(\frac{1}{6.8}, \frac{1}{5.8}, \frac{1}{4.2}, \frac{1}{3.2}; 0.8, 0.8\right)\right.$$

$$\otimes \left(\frac{1}{7}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}; 1, 1\right)\left(\frac{1}{6.8}, \frac{1}{5.8}, \frac{1}{4.2}, \frac{1}{3.2}; 0.8, 0.8\right)$$

$$\otimes \left(\frac{1}{5}, \frac{1}{4}, \frac{1}{2}, 1; 1, 1\right)\left(\frac{1}{4.8}, \frac{1}{3.8}, \frac{1}{2.2}, \frac{1}{1.2}; 0.8, 0.8\right)$$

$$\otimes \left(\frac{1}{7}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}; 1, 1\right)\left(\frac{1}{6.8}, \frac{1}{5.8}, \frac{1}{4.2}, \frac{1}{3.2}; 0.8, 0.8\right)$$

$$\left.\otimes \left(\frac{1}{7}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}; 1, 1\right)\left(\frac{1}{6.8}, \frac{1}{5.8}, \frac{1}{4.2}, \frac{1}{3.2}; 0.8, 0.8\right)^{1/5}\right)$$

$$= (0.153, 0.181, 0.287, 0.415; 1, 1)(0.158, 0.188, 0.271, 0.38; 1, 1).$$

The other matrix elements are attained by the same computational procedure. Therefore, as an example, the pairwise comparison matrix of the five experts for the evaluation dimensions are constructed as in Table 12.7.

According to the fuzzy pairwise comparison matrix, the fuzzy weights of dimensions are obtained by the following computational procedures:

$$\tilde{\tilde{r}}_1 = \left(\tilde{\tilde{a}}_{11} \otimes \tilde{\tilde{a}}_{12} \otimes \tilde{\tilde{a}}_{13} \otimes \tilde{\tilde{a}}_{14}\right)^{1/4}$$

$$= (\ (1, 1, 1, 1; 1, 1)\ (1, 1, 1, 1; 1, 1)$$

$$\otimes (0.15, 0.18, 0.29, 0.41; 1, 1)\ (0.16, 0.19, 0.27, 0.38; 0.8, 0.8)$$

$$\otimes (0.16, 0.19, 0.33, 0.52; 1, 1)(0, 17, 0.2, 0.31, 0.46; 0.8, 0.8)$$

$$\otimes (0.12, 0.13, 0.17, 0.21; 1, 1)(0.12, 0.13, 0.16, 0.2; 0.8, 0.8)\ ^{1/4}$$

$$= (0.23, 0.26, 0.36, 0.46; 1, 1)(0.24, 0.27, 0.34, 0.43; 0.8, 0.8).$$

Table 12.7 Fuzzy pairwise comparison matrix for the factors.

	F1	F2	F3	F4
F1	(1,1,1,1;1,1) (1,1,1,1;1,1)	(0.15, 0.18,0.29,0.41;1,1) (0.16,0.19,0.27,0.38;0.8,0.8)	(0.16,0.19,0.33,0.52;1,1) (0,17,0.2,0.31,0.46;0.8,0.8)	(0.12,0.13,0.17,0.21;1,1) (0.12,0.13,0.16,0.2;0.8,0.8)
F2	(2.41,3.48,5.53,6.54;1,1) (2.63,3.69,5.33,6.34;0.8,0.8)	(1,1,1,1;1,1) (1,1,1,1;1,1)	(0.72,1.32,2.64,3.62;1,1) (0.84,1.44,2.48,3.38;0.8,0.8)	(0.17,0.2,0.35,0.58;1,1) (0.17,0.21,0.32,0.51;0.8,0.8)
F3	(1.93,3.03,5.1,6.12;1,1) (2.16,3.24,4.89,5.92;0.8,0.8)	(0.28,0.38,0.76,1.38;1,1) (0.29,0.4,0.69,0.18;0.8,0.8)	(1,1,1,1;1,1) (1,1,1,1;1,1)	(0,15,0.18,0.3,0.47;1,1) (0.16,0.19,0.28,0.42;0.8,0.8)
F4	(4.83,5.86,7.73,8.56;1,1) (5.04,6.06,7.53,8.39;0.8,0.8)	(1,72,2.86,4.98,6.01;1,1) (1.96,3.08,4.77,5.81;0.8,0.8)	(2.14,3.29,5.4,6.43;1,1) (2.38,3.5,5.2,6.23;0.8,0.8)	(1,1,1,1;1,1) (1,1,1,1;1,1)

In the same way, we can obtain the remaining $\tilde{\tilde{r}}_i$ as follows:

$$\tilde{\tilde{r}}_2 = (0.73, 0.98, 1.5, 1.93; 1, 1)(0.79, 1.03, 1.44, 1.82; 0.8, 0.8)$$
$$\tilde{\tilde{r}}_3 = (0.54, 0.68, 1.04, 1.41; 1, 1)(0.57, 0.71, 0.99, 1.31; 0.8, 0.8)$$
$$\tilde{\tilde{r}}_4 = (2.05, 2.72, 3.8, 4.27; 1, 1)(2.2, 2.85, 3.7, 4.18; 0.8, 0.8).$$

The fuzzy weights are computed by normalization as follows:

$$\begin{aligned}
\tilde{\tilde{w}}_1 &= \tilde{\tilde{r}}_1 \otimes \left[\tilde{\tilde{r}}_1 \oplus \tilde{\tilde{r}}_2 \oplus \tilde{\tilde{r}}_3 \oplus \tilde{\tilde{r}}_4\right]^{-1} \\
&= (0.23, 0.26, 0.36, 0.46; 1, 1)\ (0.24, 0.27, 0.34, 0.43; 0.8, 0.8) \\
&\otimes [(0.23, 0.26, 0.36, 0.46; 1, 1)\ (0.24, 0.27, 0.34, 0.43; 0.8, 0.8) \\
&\oplus (0.73, 0.98, 1.5, 1.93; 1, 1)\ (0.79, 1.03, 1.44, 1.82; 0.8, 0.8) \\
&\oplus (0.54, 0.68, 1.04, 1.41; 1, 1)(0.57, 0.71, 0.99, 1.31; 0.8, 0.8) \\
&\oplus (2.05, 2.72, 3.8, 4.27; 1, 1)\ (2.2, 2.85, 3.7, 4.18; 0.8, 0.8)]^{-1} \\
&= (0.029, 0.039, 0.077, 0.129; 1; 1)(0.031, 0.041, 0.071, 0.114; 0.8, 0.8).
\end{aligned}$$

The remaining $\tilde{\tilde{w}}_i$ are obtained as follows:

$$\tilde{\tilde{w}}_2 = (0.091, 0.146, 0.324, 0.542; 1, 1)(0.102, 0.159, 0.297, 0.48; 0.8, 0.8)$$
$$\tilde{\tilde{w}}_3 = (0.067, 0.101, 0.224, 0.396; 1, 1)(0.073, 0.109, 0.205, 0.345; 0.8, 0.8)$$
$$\tilde{\tilde{w}}_4 = (0.255, 0.407, 0.818, 1.2; 1, 1)(0.285, 0.44, 0.763, 1.101; 0.8, 0.8).$$

Local weights and global weights for factors and criteria are given in Table 12.8. Global weights of criteria are calculated using local weights of each factor.

The costs factor is determined as the most important factor for the selection of the best location of warehouses, whereas the security factor is the least important one. The *quality and reliability of modes of transportation* criterion is defined as the most important among all criteria of warehouse location selection. The ranking of the criteria starting from the most important criterion is found to be: *quality and reliability of modes of transportation, transportation costs*, *potential of having natural disasters*, *proximity to customers and suppliers*, *proximity to major highways*, *proximity to support services*, *cost of land and buildings*, *existence of modes of transportation* and *construction costs*.

The weights for the three alternatives according to each criterion are computed by following the same calculation steps and the weights of the alternatives according to each criterion are found as in Table 12.7. The calculation of global weight of an alternative starts with building the pairwise comparison matrix of alternatives for each criterion. Then the global weight of each alternative for each criterion is multiplied by the global weight of that criterion. The sum of those products gives the global weight of the considered alternative.

Table 12.9 shows that Bursa is the best location for the warehouse of the chemical company among the alternatives.

Table 12.8 Local and global weights of factors and criteria.

	Local weights	Global weights	Defined weights	Standardized weights	Rankings	
					Factors	Criteria
F1	**(0.029, 0.039, 0.077, 0.129; 1, 1) (0.031, 0.041, 0.071, 0.114; 0.8, 0.8)**		**0.068**	**0.056**	4	
C1	(0.454, 0.63, 1.036, 1.366; 1, 1) (0.491, 0.665, 0.986, 1.284; 0.8, 0.8	(0.013, 0.024, 0.08, 0.176; 1,1) (0.015, 0.027, 0.07, 0.146; 0.8,0.8)	0.072	0.381		6
C2	(0.122, 0.15, 0.246, 0.368; 1, 1) (0.127, 0.156, 0.231, 0.333; 0.8, 0.8	(0.004, 0.006, 0.019, 0.048; 1, 1) (0.004, 0.006, 0.016, 0.038; 0.8, 0.8)	0.117	0.619		3
F2	**(0.091, 0.146, 0.324, 0.542; 1, 1) (0.102, 0.159, 0.297, 0.48; 0.8, 0.8)**		**0.273**	**0.227**	2	
C3	(0.471, 0.607, 0.942, 1.215; 1, 1) (0.498, 0.635, 0.901, 1.148; 0.8, 0.8	(0.043, 0.089, 0.305, 0.658; 1, 1) (0.051, 0.101, 0.268, 0.551; 0.8, 0.8)	0.269	0.606		4
C4	(0.152, 0.196, 0.304, 0.391; 1, 1) (0.161, 0.205, 0.29, 0.37; 0.8, 0.8	(0.014, 0.029, 0.098, 0.212; 1, 1) (0.016, 0.032, 0.086, 0.178; 0.8, 0.8)	0.175	0.394		5
F3	**(0.067, 0.101, 0.224, 0.396; 1, 1) (0.073, 0.109, 0.205, 0.345; 0.8, 0.8)**		**0.195**	**0.162**	3	
C5	(0.205, 0.307, 0.579, 0.879; 1, 1) (0.226, 0.327, 0.541, 0.795; 0.8, 0.8	(0.014, 0.031, 0.13, 0.349; 1, 1) (0.017, 0.036, 0.111, 0.275; 0.8, 0.8)	0.128	0.233		8
C6	(0.278, 0.421, 0.795, 1.192; 1, 1) (0.307, 0.451, 0.744, 1.082; 0.8, 0.8)	(0.018, 0.043, 0.178, 0.473; 1, 1) (0.022, 0.049, 0.152, 0.374; 0.8, 0.8)	0.422	0.767		1
F4	**(0.255, 0.407, 0.818, 1.2; 1, 1) (0.285, 0.44, 0.763, 1.101; 0.8, 0.8)**		**0.665**	**0.554**	1	
C7	(0.333, 0.499, 0.921, 1.295; 1, 1) (0.366, 0.534, 0.866, 1.2; 0.8, 0.8)	(0.085, 0.203, 0.753, 1.554; 1, 1) (0.104, 0.235, 0.66, 1.321; 0.8, 0.8)	0.638	0.629		2
C8	(0.106, 0.162, 0.334, 0.544; 1, 1) (0.117, 0.174, 0.309, 0.485; 0.8, 0.8)	(0.027, 0.066, 0.274, 0.652; 1, 1) (0.033, 0.077, 0.236, 0.534; 0.8, 0.8)	0.,272	0.269		7
C9	(0.05, 0.065, 0.122, 0.205; 1, 1) (0.053, 0.069, 0.113, 0.181; 0.8, 0.8)	(0.013, 0.026, 0.1, 0.246; 1,1) (0.015, 0.03, 0.086, 0.199; 0.8, 0.8)	0.104	0.102		9

Table 12.9 The local weights of the alternatives according to each criterion.

Locations	Global weights	Defined weights	Standardized weights	Rank
Bursa	(0.384,1.016,4.437,10.514;1,1)	4.007	0.523	1
	(0.484,1.192,3.819,8.618;0.8,0.8)			
Tekirdag	(0.337,0.898,3.994,9.688;1,1)	2.162	0.283	2
	(0.425,1.055,3.428,7.889;0.8,0.8)			
Balikesir	(0.213,0.551,2.513,6.432;1,1)	1.481	0.194	3
	(0.266,0.647,2.143,5.161;0.8,0.8)			

12.8 Conclusion

Type-1 fuzzy sets are somewhat problematic in defining membership functions since it is not possible to model uncertainty and imprecision sufficiently. Type-2 fuzzy sets capture this problem by incorporating the footprint of uncertainty into type-1 fuzzy sets. Fuzzy AHP based on type-2 fuzzy sets has been developed for the first time in this chapter. A linguistic scale has been also developed to be used in the proposed fuzzy AHP method. Thus a flexible definition opportunity for decision makers has been provided.

A warehouse selection problem has been investigated in this chapter. First, the warehouse selection criteria have been defined by expert opinions. After the weights of the selection factors and criteria have been determined, a warehouse location has been determined for a chemical company in Turkey by using these weights. The costs factor has been found to be the most important factor whereas the security is the least important factor. *Quality and reliability of modes of transportation* has been found to be the most important criterion which does not belong to the most important factor.

For further research, this proposed fuzzy AHP method is expected to be used in various MCDM problems such as supplier selection, project selection, and technology selection.

References

Almeida AT and Vetschera R (2012) A note on scale transformations in the PROMETHEE V method. *European Journal of Operational Research* **129**(1), 198–200.

Behzadian M, Kazemzadeh RB, Albadvi A and Aghdasi M (2010) PROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research* **200**(1), 198–215.

Birgn S, Kahraman C and Glen KG (2010) Fuzzy productivity measurement in production systems. In *Production Engineering and Management Under Fuzziness* (eds Kahraman C and Yavuz M). Springer, Berlin, pp. 417–430.

Boender CGE, de Graan JG and Lootsma FA (1989) Multicriteria decision analysis with fuzzy pairwise comparisons. *Fuzzy Sets and Systems* **29**, 133–143.

Bojković N, Anić I and Pejčić-Tarle S (2010) One solution for cross-country transport-sustainability evaluation using a modified ELECTRE method. *Ecological Economics* **69**(5), 1176–1186.

Bozbura FT, Beskse A and Kahraman C (2007) Prioritization of human capital measurement indicators using fuzzy AHP. *Expert Systems with Applications* **32**(4), 1100–1112.

Bozdag CE, Kahraman C and Ruan D (2003) Fuzzy group decision making for selection among computer integrated manufacturing systems. *Computers in Industry* **51**(1), 13–29.

Brans JP and Mareschal B (2005) PROMETHEE methods. In *Multiple Criteria Decision Analysis-State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, Boston, pp. 163–195.

Brito AJ and Almeida AT (2012) Modeling a multi-attribute utility news vendor with partial backlogging. *European Journal of Operational Research* **220**(3), 820–830.

Brito AJ, Almeida AT and Mota CMM (2010) A multi criteria model for risk sorting of natural gas pipelines based on ELECTRE TRI integrating utility theory. *European Journal of Operational Research* **200**(3), 812–821.

Buckley JJ (1985) Fuzzy hierarchical analysis. *Fuzzy Sets and Systems* **17**, 233–247.

Bulut E, Duru O, Keçeci T and Yoshida S (2012) Use of consistency index, expert prioritization and direct numerical inputs for generic fuzzy-AHP modeling: A process model for shipping asset management. *Expert Systems with Applications* **39**(2), 1911–1923.

Büyüközkan G and Çifçi G (2012) A novel hybrid MCDM approach based on fuzzy DEMATEL, fuzzy ANP and fuzzy TOPSIS to evaluate green suppliers. *Expert Systems with Applications* **39**(3), 3000–3011.

Büyüközkan G, Kahraman C and Ruan D (2004) A fuzzy multicriteria decision approach for software development strategy selection. *International Journal of General Systems* **33**, 259–280.

Celik M, Cebi S, Kahraman C and Er DI (2009) Application of axiomatic design and TOPSIS methodologies under fuzzy environment for proposing competitive strategies on Turkish container ports in maritime transportation network. *Expert Systems with Applications* **36**(3), 4541–4557.

Chan FTS, Chan MH and Tang NKH (2000) Evaluation methodologies for technology selection. *Journal of Materials Processing Technology* **107**, 330–337.

Chang DY (1996) Applications of the extent analysis method on fuzzy AHP. *European Journal of Operational Research* **95**, 649–655.

Chen SH and Hsieh CH (1999) Graded mean integration representation of generalized fuzzy number. *Journal of the Chinese Fuzzy System Association* **5**, 1–7.

Chen SJ and Hwang CL (1991) *Fuzzy Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, Berlin.

Chen SM and Lee LW (2010) Fuzzy multiple attributes group decision-making based on the interval type-2 TOPSIS method. *Expert Systems with Applications* **37**, 2790–2798.

Chen TY (2012) Comparative analysis of SAW and TOPSIS based on interval-valued fuzzy sets: Discussions on score functions and weight constraints. *Expert Systems with Applications* **39**(2), 1848–1861.

Cheng CH (1996) Evaluating naval tactical missile systems by fuzzy AHP based on the grade value of membership function. *European Journal of Operational Research* **96**(2), 343–350.

Cheng CH, Yang KL and Hwang CL (1999) Evaluating attack helicopters by AHP based on linguistic variable weight. *European Journal of Operational Research* **116**(2), 423–435.

Dong X and Hayes CC (2012) Uncertainty visualizations: Helping decision makers become more aware of uncertainty and its implications. *Journal of Cognitive Engineering and Decision Making* **6**(1), 30–56.

Duckstein L and Opricovic S (1980) Multiobjective optimization in river basin development. *Water Resources Research* **16**(1), 14–20.

Fishburn PC (1967) Additive utilities with incomplete product set: Applications to priorities and assignments. *Operations Research* **15**, 537–542.

Gungor Z, Serhatlioğlu G and Kesen SE (2009) A fuzzy AHP approach to personnel selection problem. *Applied Soft Computing* **9**(2), 641–646.

Halouani N, Chabchoub H and Martel JM (2009) PROMETHEE-MD-2T method for project selection. *European Journal of Operational Research* **195**(3), 841–849.

Haq AN and Kannan G (2006) Fuzzy analytical hierarchy process for evaluating and selecting a vendor in a supply chain model. *International Journal of Advanced Manufacturing Technology* **29**, 826–835.

Hatami-Marbini A and Tavana T (2011) An extension of the ELECTRE I method for group decision-making under a fuzzy environment. *Omega* **39**(4), 373–386.

He Y and Huang RH (2008) Risk attributes theory: Decision making under risk. *European Journal of Operational Research* **186**(1), 243–260.

Hsieh TY, Lu ST and Tzeng GH (2004) Fuzzy MCDM approach for planning and design tenders selection in public office buildings. *International Journal of Project Management* **22**, 573–584.

Javanbarg MB, Scawthorn C, Kiyono J and Shahbodaghkhan B (2012) Fuzzy AHP-based multicriteria decision making systems using particle swarm optimization. *Expert Systems with Applications* **39**(1), 960–966.

Kahraman C, Beskese A and Kaya İ (2010) Selection among ERP outsourcing alternatives using a fuzzy multi-criteria decision making methodology. *International Journal of Production Research* **48**(2), 547–566.

Kahraman C, Cebeci U and Ruan D (2004) Multi-attribute comparison of catering service companies using fuzzy AHP: The case of Turkey. *International Journal of Production Economics* **87**, 171–184.

Kahraman C, Gülbay M and Kabak Ö (2006) Application of fuzzy sets in industrial engineering: A topical classification. In *Fuzzy Applications in Industrial Engineering* (ed. Kahraman C). Springer, Berlin, pp. 1–55.

Kahraman C and Kaya İ (2010a) Fuzzy acceptance sampling plans. In *Production Engineering and Management under Fuzziness* (eds Kahraman C and Yavuz M). Springer, Berlin, pp. 457–481.

Kahraman C and Kaya İ (2010b) A fuzzy multicriteria methodology for selection among energy alternatives. *Expert Systems with Applications* **37**(9), 6270–6281.

Kahraman C, Ruan D and Doğan I (2003) Fuzzy group decision-making for facility location selection. *Information Sciences* **157**, 135–153.

Kahraman C, Ulukan Z and Tolga E (1998) A fuzzy weighted evaluation method using objective and subjective measures. In *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS '98)*, vol. 1. University of La Laguna, Tenerife, pp. 57–63.

Kang C and Feng CM (2009) Risk measurement and risk identification for BOT projects: A multi-attribute utility approach. *Mathematical and Computer Modelling* **49**(9–10), 1802–1815.

Kaya T and Kahraman C (2011a) Fuzzy multiple criteria forestry decision making based on an integrated VIKOR and AHP approach. *Expert Systems with Applications* **38**(6), 7326–7333.

Kaya T and Kahraman C (2011b) An integrated fuzzy AHP-ELECTRE methodology for environmental impact assessment. *Expert Systems with Applications* **38**(7), 8553–8562.

Kayakutlu G and Büyüközkan G (2008) Assessing knowledge-based resources in a utility company: Identify and prioritise the balancing factors. *Energy* **33**, 1027–1037.

Krohling RA and Campanharo VC (2011) Fuzzy TOPSIS for group decision making: A case study for accidents with oil spill in the sea. *Expert Systems with Applications* **38**(4), 4190–4197.

Kulak O and Kahraman C (2005) Fuzzy multi-attribute selection among transportation companies using axiomatic design and analytic hierarchy process. *Information Sciences* **170**, 191–210.

Kutlu AC and Ekmekçioğlu M (2012) Fuzzy failure modes and effects analysis by using fuzzy TOPSIS-based fuzzy AHP. *Expert Systems with Applications* **39**(1), 61–67.

Lee S, Kim W, Kim YM and Oh KJ (2012) Using AHP to determine intangible priority factors for technology transfer adoption. *Expert Systems with Applications* **39**(7), 6388–6395.

Leung LC and Cao D (2000) On consistency and ranking of alternatives in fuzzy AHP. *European Journal of Operational Research* **124**, 102–113.

Li TS and Huang HH (2009) Applying TRIZ and fuzzy AHP to develop innovative design for automated manufacturing systems. *Expert Systems with Applications* **36**(4), 8302–8312.

Lin HF (2010) An application of fuzzy ahp for evaluating course website quality. *Computers & Education* **54**, 877–888.

Løkena E, Botterudb A and Holena AT (2009) Use of the equivalent attribute technique in multi-criteria planning of local energy systems. *European Journal of Operational Research* **197**(3), 1075–1083.

Malak RJ, Aughenbaugh MJ and Paredis JJC (2009) Multi-attribute utility analysis in set-based conceptual design. *Computer-Aided Design* **197**(3), 214–227.

Mateo JRSC (2012) Weighted sum method and weighted product method. In *Multi Criteria Analysis in the Renewable Energy Industry* (eds Mateo JRSC and Ramón J). Springer, Berlin, pp. 19–22.

Mendel JM and John RIB (2002) Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems* **10**(2), 117–127.

Mendel JM, John RIB and Liu FL (2006) Interval type-2 fuzzy logical systems made simple. *IEEE Transactions on Fuzzy Systems* **14**(6), 808–821.

Montazer AG, Saremi HQ and Ramezani M (2009) Design a new mixed expert decision aiding system using fuzzy ELECTRE III method for vendor selection. *Expert Systems with Applications* **36**(8), 10837–10847.

Rajput HC, Milani AS and Labun A (2011) Including time dependency and ANOVA in decision-making using the revised fuzzy AHP: A case study on wafer fabrication process selection. *Applied Soft Computing* **11**(8), 5099–5109.

Ramesha S, Kannanb S and Baskarc S (2012) Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Applied Soft Computing* **12**, 741–753.

Rong C, Takashi K and Wang J (2003) Enterprise waste evaluation using the analytic hierarchy process and fuzzy set theory. *Production Planning and Control* **14**(1), 90–103.

Roy B (2005) Paradigms and challanges. In *Multiple Criteria Decision Analysis-State of the Art Surveys* (eds Figueira J, Greco S and Ehrgott M). Springer, Boston, pp. 3–24.

Saaty TL (1980) *The Analytic Hierarchy Process*. McGraw-Hill, New York.

Saaty TL (1990) How to make a decision: The analytic hierarchy process. *European Journal of Operations Research* **48**, 9–26.

Saaty TL (2008) *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. RWS Publications, Pittsburgh.

Stam A, Minghe S and Haines M (1996) Artificial neural network representations for hierarchical preference structures. *Computers and Operations Research* **23**(12), 1191–1201.

Torlak G, Sevkli M, Sanal M and S. Z (2011) Analyzing business competition by using fuzzy TOPSIS method: An example of Turkish domestic airline industry. *Expert Systems with Applications* **38**(4), 3396–3406.

Triantaphyllou E (2000) *Multi-criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Publishers, New York.

Tsaur SH, Chang TY and Yen CH (2002) The evaluation of airline service quality by fuzzy MCDM. *Tourism Management* **23**, 107–115.

Tseng ML, Lin YH, Chiu AS and Chen CY (2008) Fuzzy AHP-approach to TQM strategy evaluation. *Industrial Engineering and Management Systems: An International Journal* **7**(1), 34–43.

van Laarhoven PJM and Pedrycz W (1983) A fuzzy extension of Saaty's priority theory. *Fuzzy Sets and Systems* **11**, 229–241.

Vázqueza ALJ, Cantob EB and A. M (2008) Optimal design and operation of a wastewater purification system. *Mathematics and Computers in Simulation* **79**(3), 668–682.

Vetschera R and Almeida AT (2012) A PROMETHEE-based approach to portfolio selection problems. *Computers & Operations Research* **39**(5), 1010–1020.

Vincke P (1992) *Multicriteria Decision Aid*. John Wiley & Sons, Ltd, New York.

Wang M, Lin SJ and Lo YC (2010) The comparison between MAUT and PROMETHEE. In Proceedings of the 2010 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 753–757.

Wu MC and Chen TY (2011) The ELECTRE multicriteria analysis approach based on Atanassov's intuitionistic fuzzy sets. *Expert Systems with Applications* **38**(10), 12318–12327.

Yilmaz B and Dağdeviren M (2011) A combined approach for equipment selection: F-PROMETHEE method and zero-one goal programming. *Expert Systems with Applications* **38**(9), 11641–11650.

Yoon KP and Hwang CL (1995) *Multiple Attribute Decision Making: An Introduction*. Sage Publications, London.

Zadeh LA (1965) Fuzzy sets. *Information and Control* **8**(3), 338–353.

Zadeh LA (1975) The concept of a linguistic variable and its application to approximate reasoning – i. *Information and Control* **8**(3), 199–249.

Zeng J, Min A and Smith NJ (2007) Application of fuzzy based decision making methodology to construction project risk assessment. *International Journal of Project Management* **25**, 589–600.

Zhu KJ, Jing Y and Chang DY (1999) A discussion of extent analysis method and applications of fuzzy AHP. *European Journal of Operational Research* **116**, 450–456.

Zimmermann HJ (1996) *Fuzzy Set Theory and its Applications*. Kluwer, Boston.

Zolghadri M, Amrani A, Zouggar S and Girard P (2011) Power assessment as a high-level partner selection criterion for new product development projects. *International Journal of Computer Integrated Manufacturing* **24**(4), 312–327.

13

Applying genetic algorithms to optimize energy efficiency in buildings

Christina Diakaki and Evangelos Grigoroudis

Department of Production Engineering and Management, Technical University of Crete, Greece

13.1 Introduction

The energy sector faces evidently significant challenges that every day become even more acute. According to the United Nations Intergovernmental Panel on Climate Change (IPCC), unless reductions of at least 50% in global CO_2 emissions compared with the 2000 level is achieved by 2050, the aim of limiting the long-term global average temperature rise by 2.0−2.4 °C will not be achieved (International Energy Agency 2010). In addition, according to the International Energy Agency (IEA), climate change is occurring even faster than previously expected so that the 50% reduction by 2050 may be inadequate to prevent a dangerous climate change (International Energy Agency 2010).

The trends in CO_2 emissions are driven by the amount and type of energy used, and the indirect emissions associated with the production of electricity. Among the greater energy consumers is the building sector. Buildings today account for 40% of the world's energy use, while their resulting CO_2 emissions are substantially more than those of the transportation sector. Households, in particular, have a share of 29% of the total energy consumed globally, and release 21% of the total emissions (International Energy Agency

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

2008). The household sector includes all energy-using activities in apartments and houses, including space and water heating, cooling, lighting and the use of appliances.

From an energy perspective, buildings are complex systems considering the building envelope and its insulation, the space heating and cooling systems, the water heating systems, the lighting appliances and other equipment. In contrast, however, to other systems, most buildings have a long life span. This means that more than the half of the current global building stock will still be standing in 2050, while, at the same time, new buildings that will use more energy than necessary are being built every day (International Energy Agency 2010; World Business Council for Sustainable Development 2009). These also mean that most of the energy and CO_2 savings potential lies in the retrofitting and purchasing of new technologies for the existing building stock, as well as in the efficient design and establishment of improved standards for the new buildings.

Therefore, as far as the building sector is concerned, to reduce the planet's energy-related carbon footprint so as to reach the level called for by the IPCC, an aggressive reduction of energy use in new and existing buildings is necessary (World Business Council for Sustainable Development 2009). To this end, several measures, i.e., interventions are available both for the design and the operational stage of a building, when renovation or retrofit actions are needed. These measures may be distinguished in the following basic categories (Diakaki *et al*. 2008):

- Measures for the improvement of the building's envelope (addition or improvement of insulation, change of color, placement of heat-insulating door and window frames, increase of thermal mass, building shaping, superinsulated building envelopes, etc.).
- Measures for reducing the heating and cooling loads (exploitation of the principles of bioclimatic architecture, incorporation of passive heating and cooling techniques, i.e., cool coatings, control of solar gains, electrochromic glazing, etc.).
- Use of renewables (solar thermal systems, buildings' integrated photovolatics, hybrid systems, etc.).
- Use of 'intelligent' energy management, i.e., advanced sensors, energy control (zone heating and cooling) and monitoring systems.
- Measures for the improvement of the indoor comfort conditions in parallel with minimization of the energy requirements (increase of ventilation rate, use of mechanical ventilation with heat recovery, improvement of boilers and air-conditioning efficiency, use of multifunctional equipment, i.e., integrated water heating with space cooling, etc.).
- Use of energy efficient appliances and compact fluorescent lighting.

With such a variety of proposed measures, the main issue and problem is to identify those that will be proven to be the more effective and reliable in the long term. In this identification process, the decision maker (DM), meaning the civil, mechanical and environmental engineer as well as the architect involved in the building design and/or retrofit project has to compensate for environmental, energy, financial or other factors in order to reach the best possible and feasible solution; the solution, will ensure maximization of energy efficiency, while satisfying, at the same time, the needs of the building's final

user/occupant/owner. In other words, the DM faces a multiple criteria decision problem, where the search of an optimal solution is meaningless, since the criteria, which have to be satisfied are generally competitive (e.g., energy efficient solutions are more expensive than less efficient ones).

The state-of-practice approach to the previously defined problem involves initially, the performance of an energy analysis of the building under study, based on which, the DM defines several alternative scenarios, which are expected to improve its energy performance (e.g., addition of insulation, and/or change of boiler, etc.) (Krarti, 2000). These specific scenarios, which may vary according to buildings' characteristics, type, use, climatic conditions, etc., are then evaluated mainly through simulation using more or less advanced/detailed calculations. Sometimes, the DM employs complementary to simulation, decision supporting techniques, such as multicriteria decision analysis (MCDA) techniques to assist him/her in reaching the final decision, among the set of predefined and pre-evaluated alternative actions [see Kolokotsa *et al*. (2009) for a complete state-of-the art review].

When performing the aforementioned approach, the DM faces another problem too. If he/she defines only a limited number of solutions to be evaluated, there is no guarantee that the solution finally reached is the best possible. In fact, there is no guarantee that the solution finally reached is among the set of good solutions, in the sense that there might be other solutions performing better in all considered criteria. Also, the selection of a representative set of alternatives is usually a difficult problem, while the final solution is heavily affected by these predefined alternatives. In the opposite case, i.e., if the DM defines numerous solutions, the required evaluation and selection process may become extremely time-consuming and difficult to handle. As a consequence, the study is limited to a potentially large but certainly finite number of alternative scenarios and actions (i.e., solutions), when the real opportunities are enormous considering all the available improvement measures that may be employed. At the same time, the whole process, as well as the final decisions, is significantly affected by the DM's experience and knowledge. Although this experience and knowledge are certainly significant and irreplaceable elements to the whole process, it is however, necessary to develop practical tools that will assist him/her in taking into account as many feasible alternatives and decision criteria as possible, without the restrictions imposed by the predefined scenarios.

To avoid the need to prescribe and evaluate alternative scenarios, some researchers define the decision problem as a multiobjective one, either by formulating a mathematical program or by developing an energy simulation model. In both cases, genetic algorithms (GAs) are among the methods usually used to find satisfactory solutions. The GAs are used as search engines, while the energy models are used in the evaluation of the solutions proposed by the GAs. The necessity of applying GAs is justified by the size and the complexity of the examined optimization problems.

It is the aim of this chapter to review and discuss multiobjective optimization GA-based approaches to the problem of improving energy efficiency in buildings, and to demonstrate, through a simple example case study, how a GA could be applied to this problem. The case study will also allow to highlight the potential strengths and weaknesses of this approach.

The chapter is structured into four more sections. Section 13.2 provides the state-of-the-art review of the application of GAs to the multiobjective improvement of energy efficiency in buildings. Section 13.3 presents the example case study and the development

of a related multiobjective decision problem. Section 13.4 describes the development of a GA for the defined decision problem, and the solution of the problem through the developed GA. Finally, Section 13.5 summaries the conclusions and main findings of the chapter.

13.2 State-of-the-art review

A critical aspect in the design but also in the operational phase of a building, when renovation or retrofit actions are needed, is the evaluation and adjustment of the alternative measures/interventions based on a set of criteria such as energy consumption, environmental performance, investment cost, operational cost, indoor environment quality, security, social factors, etc. (Kolokotsa *et al*. 2009). In some cases, the aforementioned criteria are competitive in nature, or interrelate in a nonlinear way, making the problem of reaching a globally optimal solution generally infeasible. For this reason, a feasible intermediary solution is suggested that will satisfy the specific requirements of the building's final user/occupant/owner.

The different actions that may be considered through this process include more than 400 alternatives (Wulfinghoff 1999) and may be accomplished separately or combined in groups, thus leading to an enormous number of alternative solutions that should be taken into account. The assessment of these alternative solutions either at the design stage of a new building or at the operational stage of an existing building, for retrofit or renovation purposes, aims to identify particular measures such as insulation materials, wall structure, heating, ventilating and air conditioning (HVAC) type, etc., that are expected to lead to improved building energy and environmental performance.

As mentioned in Section 13.1, the most widely used approach for the assessment of the alternative solutions is the energy analysis of the building under study via simulation, while the final decision is sometimes assisted through MCDA techniques, which are performed upon the set of predefined and pre-evaluated alternative solutions. Relatively recently however, new approaches have emerged, which combine the power of GAs in exploring a given decision space with simulation or other energy analysis tools in an effort to confront the problem of energy efficiency in buildings in its real dimensions, which include an enormous number of alternative solutions. The approaches reported herein concern methods and tools aimed at intermediary to late stages of building design and at building renovations, when, i.e., the solution space is more concrete.

One of the first efforts to apply GAs to the problem of improving energy efficiency in buildings is reported by Caldas and Norford (2003). Caldas and Norford (2002) developed initially a tool for the optimal sizing and placement of windows in terms of thermal and lighting performance in the building. Their GA generated possible design solutions, which were then evaluated using DOE 2.1 (Simulation Research Group 1993) thermal analysis software. In particular, they used the so-called micro-GA (Krishnakumar 1989). Later on, Caldas and Norford (2003) evaluated the optimal trade-offs between reductions in energy consumption during building operation and initial costs due to construction materials, as well as employing GAs to alter the building form to optimize the trade-off of lighting and heating energy.

Caldas (2008) combined also a standard GA with the DOE 2.1 software to develop GENE_ARCH, an evolution-based generative design system. GENE_ARCH may be used in two types of problems: problems where the building geometry is fixed and

optimization acts upon specific aspects of defined architectural solutions (such as façade design, window sizing, shading systems or construction materials); and problems where the building geometry itself is being generated by the system, a much more complex issue. For the optimization, criteria such as the energy spent for heating, cooling, ventilation and artificial lighting, the cost of construction materials and the greenhouse gas emissions associated with the embodied energy of these materials were considered.

Wright and Farmani (2001a,b) and Wright *et al*. (2002) report on the implementation of a multiobjective genetic algorithm (MOGA) approach (Fonseca and Fleming 1995, 1998). The proposed approach was initially developed to optimize the design variables of HVAC, and was then extended to optimize additional variables including the operation variables of HVAC, the building weight and the glazing type and area. The MOGA was used to explore the trade-offs between operating costs and thermal comfort and between operating and capital costs. The performance of each candidate design solution was evaluated using a thermal performance simulation of the building and the associated HVAC system (Ren and Wright 1998).

The aforementioned MOGA approach (Fonseca and Fleming 1995, 1998) coupled with a simulation program based on the ASHRAE toolkit for building load calculations (Pedersen *et al*. 2000) was also employed by Wang *et al*. (2005) to define orientation, aspect ratio, window type, window-to-wall ratio for each building façade, wall and roof types (i.e., sequences of layers) and wall and roof layers (i.e., materials) so as to optimize the life cycle cost and environmental impact of buildings. Wang *et al*. (2005) employed an improved version of the traditional GA, called a structured GA (Dasgupta and McGregor 1993).

Another interesting approach has been proposed by Verbeeck and Hens (2007), who employed a MATLAB developed GA, coupled with TRNSYS (TRNSYS 2005) simulations for fitness calculations, and input data for ventilation and infiltration calculated by the COMIS air flow model (COMIS 2003). Their methodology performs a two-stage optimization. In the first stage, only envelope-related energy saving measures are considered (such as roof, attic floor, façade and ground floor, glazing type and area, sun shading, air tightness and natural ventilation) in order to optimize the net heat demand, the investment cost and the environmental impact. For the optimal building-envelope measures defined in the first stage, in a second stage, system-related measures for distribution, emission, production and storage of heat, systems for local electricity production and control systems are considered in order to reduce yearly energy consumption and maintenance costs.

A similar approach is also described by Chantrelle *et al*. (2011), who use the Nondominated Sorting Genetic Algorithm-II (NSGA-II) developed by Deb (2001, 2005) coupled with TRNSYS, and economic and environmental databases in order to develop the multicriteria tool MultiOpt. MultiOpt, in contrast to the previously described approaches, aims at building renovation evaluations considering energy consumption, cost, life-cycle environmental impact and thermal comfort. NSGA-II is recognised as one of the most efficient multiobjective evolutionary algorithms (MOEAs), a specific class of GAs based on Pareto dominance. In the GA approach of Chantrelle *et al*. (2011), an individual represents the result of a renovation operation carried out on a building, which may refer to either a building control strategy (such as cooling and shading control) or the building envelope characteristics (such as external wall type, roof type, ground floor type, intermediate floor type, internal partition wall type, and window type).

A MOGA has also been employed by Chen and Gao (2011) for the optimization of building energy performance. In this study, a building information model (BIM) was built to provide design data, such as building form and space layout, and site and building orientation to IES<VE> (Integrated Environmental Solutions 2011), a building energy simulation software. This way, energy performance of design options was evaluated, and the optimal settings of the design parameters were then obtained using the MOGA.

A final approach in the same context, i.e., coupling of a GA with simulation or other evaluation methods, has been reported by Hamdy *et al*. (2011). This study combines a GA developed using available MATLAB tools (Hamdy *et al*. 2009, 2011) with the IDA ICE (Sahlin *et al*. 2004) building performance simulation program under the MATLAB environment. The aim of the study was to achieve low-emission, cost-effective design solutions for the insulation thickness of the external wall, roof and floor, the thermal conductivity of the windows and the type of heat recovery. Therefore CO_2 emissions related to heating energy, and the investment cost related to the suggested design variables were selected as the two objective functions to be minimized. The employed GA was a combination of two previously developed GAs (Hamdy *et al*. 2009):

- The PR-GA (preparation process and GA), which is the MATLAB 2008a GA, modified appropriately in order to deal with both discrete and continuous variables, and combined with the FMINCON, the deterministic optimization solver of MATLAB in order to provide it with a good collection of individuals as an initial population.
- The GA-RF (GA with refine process), which is a hybrid optimization solver that after the use of the PR-GA, utilizes FGOALATTAIN, the MATLAB multiobjective optimization solver, in order to improve the quality and diversity of the obtained results.

As can be seen from the above review, to avoid the need to prescribe alternative solutions, GAs are employed to search the decision space, while simulation or other energy analysis tools are still used to evaluate the solutions proposed by the GAs. This approach however, is still computationally expensive, since the time associated with optimization can become prohibitively high due to the usually large number of simulations that need to be performed.

To overcome this problem, Magnier and Haghighat (2010) proposed the GAINN approach. GAINN uses first a simulation-based artificial neural network (ANN) to characterize building behavior, and then combines this ANN with a multiobjective GA for optimization. Magnier and Haghighat (2010) use the NSGA-II algorithm and TRNSYS, also employed by Chantrelle *et al*. (2011), though in a different way. In this study an ANN is first trained and validated through TRNSYS simulations of the building under study, and then used inside the NSGA-II to evaluate potential solutions. Design variables in the study were building envelope- and HVAC system-related measures, while the considered decision criteria were thermal comfort and energy consumption.

Table 13.1 summarizes the findings of the above review, while the next sections demonstrate, through a simple example, the development of a GA for the multiobjective optimization of energy in buildings without any coupling with simulation or other energy analysis tools, an approach which may lead to a significant reduction of the necessary computational effort compared with all the previously reviewed GA approaches.

Table 13.1 Summary of state-of-the-art review.

Decision variables	Decision criteria	Approach	Reference
Sizing and placement of windows	Thermal and lighting performance	Micro-GA coupled with DOE 2.1 software	Caldas and Norford (2003)
Design and operation variables of HVAC, building weight, and glazing type and area	Operating costs and thermal comfort or operating and capital costs	MOGA coupled with thermal performance simulation	Wright and Farmani (2001a, b), Wright *et al*. (2002)
Construction materials or building form	Energy consumption and initial costs or lighting and heating energy	Micro-GA coupled with DOE 2.1 software	Caldas and Norford (2003)
Orientation, aspect ratio, window type, window-to-wall ratio for each building façade, wall and roof types and materials	Life cycle cost and environmental impact	Structured GA coupled with an ASHRAE toolkit for building load calculations	Wang *et al*. (2005)
Measures relating to the envelope and the heating, local electricity production and control systems	Net heat demand, investment cost, environmental impact, yearly energy consumption and maintenance costs	MATLAB developed GA, coupled with TRSNYS software and COMIS model	Verbeeck and Hens (2007)
Aspects of defined architectural solutions or construction materials or building geometry	Heating, cooling, ventilation and artificial lighting energy consumption, construction materials cost and greenhouse gas emissions	Standard GA coupled with the DOE 2.1 thermal analysis software	Caldas (2008)

(*continued overleaf*)

Table 13.1 (*Continued*)

Decision variables	Decision criteria	Approach	Reference
Building envelope- and HVAC system-related measures	Thermal comfort and energy consumption	NSGA-II coupled with a TRNSYS-trained ANN	Magnier and Haghighat (2010)
Building control strategy or envelope characteristics	Energy consumption, cost, life-cycle environmental impact and thermal comfort	NSGA-II coupled with TRNSYS software, and economic and environmental databases	Chantrelle *et al*. (2011)
Design options	Energy performance and cost	MOGA coupled with BIM and IES<VE> software	Chen and Gao (2011)
Insulation thickness of external wall, roof and floor, thermal conductivity of windows and type of heat recovery	Heating-related CO_2 and investment cost	MATLAB GA coupled with the IDA ICE software	Hamdy *et al*. (2011)

13.3 An example case study

13.3.1 Basic principles and problem definition

The development of a GA-based approach for the improvement of energy efficiency in buildings considering multiple criteria requires:

- the definition of decision variables, discrete and/or continuous, to reflect the total set of alternative measures that are available and will be considered;
- the identification and formulation into appropriate linear and/or nonlinear mathematical expressions of the objectives to be achieved;
- the delimitation of the set of feasible solutions through the identification of linear and/or nonlinear constraints concerning either the decision variables and their intermediary relations or the objectives of the problem (natural and logical constraints may also be considered as necessary);
- the development of a GA capable of handling the defined variables, objectives and constraints.

The example developed herein, aims to provide an insight in this process and highlight any potential strengths and weaknesses of the approach. To this end, a simple case is considered, which has been adapted from a previous study (Diakaki *et al*. 2008). The problem concerns the construction of the simple building displayed in Figure 13.1. The dimensions of the building envelope assume a floor and ceiling area of 100 m^2, 2 walls of area 24 m^2, 2 walls of area 30 m^2, and a door and window area both of 6 m^2. Figure 13.2 displays the structure of the walls that is assumed for this building. The structure consists of the following sequence of layers from outside to inside: concrete, insulation and gypsum board.

The decisions regarding the building under study concern the window type and the material and thickness for the wall's insulation layer. For simplicity reasons, all other elements of the building envelope have been assumed defined, thus left outside the study and consequently the decision model.

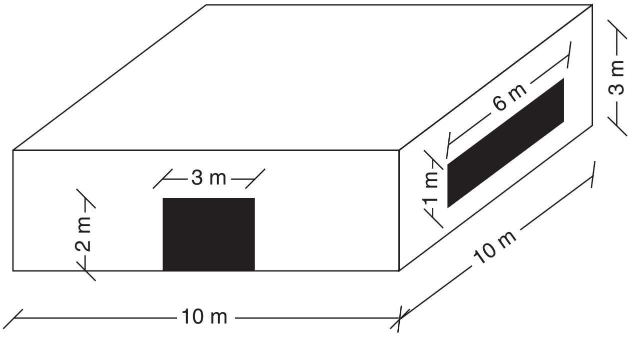

Figure 13.1 The building under study. Reprinted from Diakaki *et al*. 2008 with permission from Elsevier.

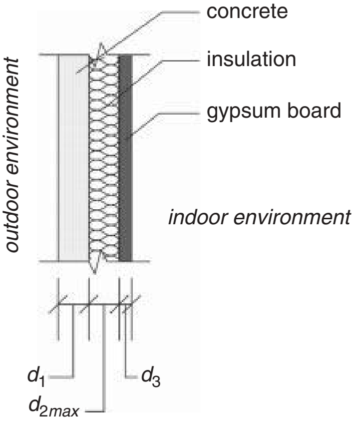

Figure 13.2 Construction layers of the building's walls. Reprinted from Diakaki *et al*. 2008 with permission from Elsevier.

The pursued goals for the building are to reduce the cost for the acquisition of materials, which corresponds to the initial investment cost, and to increase the resulting energy savings, which reflect the future operational costs of the building. These goals are competitive, since materials, which improve the energy savings, are usually more expensive.

13.3.2 Decision variables

As mentioned earlier, the decisions considered in this example case study concern three choices. For these three choices, two types of decision variables are defined, binary variables to reflect the alternative choices regarding the window type and the wall's insulation material, and a continuous variable to correspond to the thickness of the insulation layer of the wall. The defined decision variables with their associated constraints are described below.

Assuming that I different types of windows may be considered, binary variables x_{1i} with $i = 1, 2, \ldots, I$ are defined as follows:

$$x_{1i} = \begin{cases} 1, & \text{if window of type } i \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{13.1}$$

Since only one window type will be selected, the following constraint holds for these variables

$$\sum_{i=1}^{I} x_{1i} = 1 \tag{13.2}$$

Assuming that J different materials may be considered for insulation, binary variables x_{2j} with $j = 1, 2, \ldots, J$ are defined as follows:

$$x_{2j} = \begin{cases} 1, & \text{if insulation material of type } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \tag{13.3}$$

Since, also, only one insulation material will be selected, the following constraint holds for these latter decision variables

$$\sum_{j=1}^{J} x_{2j} = 1 \tag{13.4}$$

Finally, as far as the thickness of the wall's insulation layer is concerned, it is denoted by the continuous variable x_2 (in m^2) for which the following constraint holds

$$d_{2min} \le x_2 \le d_{2max} \tag{13.5}$$

where $d_{2\,\mathrm{min}}$ (in m^2) represents the minimum required insulation thickness, and d_{2max} (in m^2) the maximum permissible value designated by the available space (Figure 13.2).

13.3.3 Decision criteria

As mentioned earlier, the aim of the study is twofold and concerns the reduction of the initial investment cost and the increase of the energy savings.

The initial investment cost concerns only the cost for the acquisition of materials. Therefore, the total cost C may be simply obtained, by adding the cost of the windows and the cost of the wall's insulation material. Assuming that A_{WIN} is the window surface (in m^2), C_{1i} is the cost (in euros/m^2) for window type i with $i = 1, \ldots, I$, A_{WAL} is the surface of the walls to be insulated (in m^2), and C_{2j} is the cost (in euros/m^3) for insulation material j with $j = 1, \ldots, J$, the total initial investment cost is obtained through:

$$C = A_{WIN} \sum_{i=1}^{I} C_{1i} x_{1i} + A_{WAL} x_2 \sum_{j=1}^{J} C_{2j} x_{2j} \tag{13.6}$$

Concerning the second aim of the study, that is the increase of energy savings, several options are available. Herein, this aim is approached through the choice of materials with low thermal conductivity. Therefore the corresponding decision criterion should be developed so as to allow for such a choice. An appropriate decision criterion in this respect is the building load coefficient (BLC).

Generally, the BLC is calculated according to the equation $BLC = \sum_{com} A_{com} U_{com} b_{com}$ (Krarti 2000), where com is a building envelope component (with one unique U-value), and A_{com} and U_{com} are the surface area (m^2) and the thermal transmittance [W/(m^2 K)] of the component, respectively. Moreover, b_{com} is a temperature correction factor that is equal to 1 for components facing the outside air such as doors, windows, outside walls and roofs, and lowers, down to 0, for surfaces facing earth or unheated spaces such as ground, floors and cellar of crawl spaces (Berben *et al*. 2004). Applying this equation to the building under study, and taking into account that the considered building components, i.e., the windows and the walls all face the outside air, thus, their temperature correction factors are equal to 1, the following formula results

$$BLC = A_{WIN} U_{WIN} + A_{WAL} U_{WAL} \tag{13.7}$$

where U_{WIN} and U_{WAL} are the thermal transmittance of the window and the wall, respectively.

The thermal transmittance of the window is simply calculated through the following formula

$$U_{WIN} = \sum_{i=1}^{I} U_{1i} x_{1i} \tag{13.8}$$

where U_{1i} is the thermal transmittance of window type i, while, in the case of the walls, the necessary calculations are more complex. Assuming that the wall is constructed from several homogeneous layer parts, its thermal transmittance is calculated through the following formula (Krarti 2000):

$$U_{WAL} = \frac{1}{\sum_{l} \frac{d_l}{k_l}} \tag{13.9}$$

where d_l, and k_l are the thickness (in m) and the thermal conductivity [in W/(m K)] of layer part l, respectively, and l (where $l = 1, 2, \ldots$) is the index of the construction layer.

In the considered building, the walls are assumed to be constructed from three layers (Figure 13.2). From these three layers, the construction of two is assumed known and given, therefore, the choice concerns only the third layer that is the insulation layer (i.e., the intermediate layer in Figure 13.2). Introducing these in Equation (13.9) the following formula results

$$U_{WAL} = \frac{1}{\sum_{n=1}^{2}\left(\frac{d_n}{k_n}\right) + \frac{x_2}{k_2}} \tag{13.10}$$

where n is an index to the two known construction layers of the wall for which the thicknesses d_n as well as the thermal conductivities k_n are known, while x_2 and k_2 correspond to the thickness and the thermal conductivity, respectively, of the unidentified insulation layer, and depend upon the choice of the insulation material (i.e., x_{2j}). Therefore, the following equation results for the calculation of the thermal transmittance of the wall

$$U_{WAL} = \frac{1}{\sum_{n=1}^{2}\left(\frac{d_n}{k_n}\right) + \frac{x_2}{\sum_{j=1}^{J}\left(k_{2j}x_{2j}\right)}} \tag{13.11}$$

where k_{2j} is the thermal conductivity of insulation material j.

Introducing Equations (13.8) and (13.11) in (13.7), the following formula results that describes the load coefficient of the building under study

$$BLC = A_{WIN}\sum_{i=1}^{I}\left(U_{1i}x_{1i}\right) + \frac{A_{WAL}}{\sum_{n=1}^{2}\left(\frac{d_n}{k_n}\right) + \frac{x_2}{\sum_{j=1}^{J}\left(k_{2j}x_{2j}\right)}} \tag{13.12}$$

13.3.4 Decision model

The decision variables and criteria developed in the previous subsections, lead to the formulation of the following multiobjective decision problem:

$$[\min]\ g_1(x) = C = A_{WIN}\sum_{i=1}^{I}\left(C_{1i}x_{1i}\right) + A_{WAL}x_2\sum_{j=1}^{J}\left(C_{2j}x_{2j}\right)$$

$$[\min]\ g_2(x) = BLC = A_{WIN}\sum_{i=1}^{I}\left(U_{1i}x_{1i}\right) + \frac{A_{WAL}}{\sum_{n=1}^{2}\left(\frac{d_n}{k_n}\right) + \frac{x_2}{\sum_{j=1}^{J}\left(k_{2j}x_{2j}\right)}}$$

s.t.

$$\sum_{i=1}^{I} x_{1i} = 1 \tag{13.13}$$

$$\sum_{j=1}^{J} x_{2j} = 1$$

$$x_{1i} \in \{0, 1\} \ \forall i \in \{1, \ldots, I\}$$

$$x_{2j} \in \{0, 1\} \ \forall j \in \{1, \ldots, J\}$$

$$x_2 \in \left[d_{2min}, d_{2max}\right]$$

where the data described below have been assumed.

For the window, four types with the characteristics displayed in Table 13.2 have been assumed to be available in the market. It has also been assumed that there are four types of insulation materials available with the characteristics displayed in Table 13.3. The values of the thermal transmittance and the thermal conductivity in these two tables have been chosen from the ASHRAE database (Kreider *et al*. 2002), while the values for the costs have been set so as to reflect the cost increase that accompanies the decrease of thermal conductivity.

Moreover, considering the building dimensions, as displayed in Figure 13.1, A_{WIN} is 6 m^2, A_{WAL} is 108 m^2, the minimum required thickness d_{2min} for the insulation layer of the wall is 0 m, while the maximum permissible thickness d_{2max} is 0.10 m. Moreover, for the predefined layers 1 and 3 of the wall construction, thicknesses of 0.10 m and 0.01 m, and thermal conductivities 0.75 and 0.48 W/(m K) have been considered, respectively.

The developed decision problem (13.13) is a nonlinear mixed-integer multiobjective combinatorial optimization problem (Collette and Siarry 2004; Ehrgott 2005). Moreover, as already mentioned, the two involved criteria are competitive, since any decrease of the one, leads to an increase of the other. Table 13.4, displaying the payoff matrix when each

Table 13.2 Characteristics and data for different window types. Reprinted from Diakaki *et al*. 2008 with permission from Elsevier.

Window types	Thermal transmittance [W/(m^{2}K)]	Cost (euros/m^2)
Single pane windows	$U_{11} = 6.0$	$C_{11} = 50$
Double pane windows of 6 mm space	$U_{12} = 3.4$	$C_{12} = 100$
Double pane windows of 13 mm space	$U_{13} = 2.8$	$C_{13} = 150$
Double low-e windows	$U_{14} = 1.8$	$C_{14} = 200$

Table 13.3 Characteristics and data for different insulation materials. Reprinted from Diakaki *et al*. 2008 with permission from Elsevier.

Insulation types	Density (kg/m^3)	Thermal conductivity [W/(m K)]	Cost (euros/m^3)
Cellular glass	136.000	$k_{21} = 0.050$	$C_{21} = 50$
Expanded polystyrene molded beads	16.000	$k_{22} = 0.029$	$C_{22} = 100$
Cellular polyurethane	24.000	$k_{23} = 0.023$	$C_{23} = 150$
Polyisocyanurate	0.020	$k_{24} = 0.020$	$C_{24} = 200$

Table 13.4 Payoff matrix.

Type of solution	C (euros)	BLC (W/K)	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_2 (m)
[min] $g_1(x)$	300.00	736.54	1	0	0	0	0	0	0	0	0.0
[min] $g_2(x)$	3360.00	31.75	0	0	0	1	0	0	0	1	0.1

criterion is optimized independently from the other, demonstrates this competitiveness. When the cost criterion is optimized, low-cost decision choices are made that may, however, lead to decreased energy savings since the building load coefficient takes high values and vice versa.

The typical (widely used) approach to solve computationally heavy multiobjective optimization problems is to focus only on the estimation of the Pareto frontier. This approach has also been used in all but the Wright and Farmani (2001a, b) and Wright *et al*. (2002) studies reviewed in the previous section. However, such an approach does not take into account the DM's preferences and does not provide a solution to the examined decision problem, but rather it tries to offer a range of solutions to the DM. For this reason, in several cases, MCDA techniques (e.g., analytic hierarchy process ELECTRE) are applied to the Pareto frontier in a second stage. In this chapter, though, a different approach is adopted in order to model DM's preferences in the mathematical program.

For the solution of problem (13.13), the GA available in the Global Optimization Toolbox of MATLAB R2011b (MathWorks 2011) has been used, while to introduce the DM's preferences in the mathematical program, and lead to a single solution, rather than the provision of a range of solutions, the principles of compromise programming (Collette and Siarry 2004; Ehrgott 2005) have been used.

According to compromise programming, the multiple decision criteria are aggregated into a single criterion aiming at minimizing the distance of their values from their optimum ones. In other words, the multiobjective problem (13.13) is transformed into the following single-objective problem:

$$\begin{aligned}
&[\min]\ z = \lambda \\
&\text{s.t.} \\
&\text{all constraints of multiobjective problem (13.13)} \\
&\lambda \geq \left(g_1(x) - g_{1min}\right)\left(p_1/g_{1min}\right) \\
&\lambda \geq \left(g_2(x) - g_{2min}\right)\left(p_2/g_{2min}\right) \\
&\lambda \geq 0
\end{aligned} \tag{13.14}$$

where λ corresponds to the Tchebyshev distance, g_{1min} and g_{2min} are the optimum (minimum) values of the two criteria when optimized independently (Table 13.4), and p_1 and p_2 are the corresponding weight coefficients reflecting the relative importance of the two criteria. The weight coefficients allow the DM to express his/her preferences regarding the criteria, and must sum up to 0, i.e., $p_1 + p_2 = 1$ should hold. The problem (13.14) includes 10 decision variables, 2 of which are continuous (λ and x_2) and 8 binary (x_{1i}, $i = 1, \ldots, 4$ and x_{2j}, $j = 1, \ldots, 4$), as well as 2 linear equality and 2 nonlinear inequality constraints.

Problem (13.14) has been programmed in MATLAB, and several runs have been performed to improve the GA's results through experimentation with its different options.

During this experimentation, a DM has been assumed with equal preferences to both considered criteria, i.e., with $p_1 = p_2 = 0.5$. After selecting appropriate settings for the GA's options, the GA has been used to solve problem (13.14) for several settings of the decision problem's weight coefficients.

13.4 Development and application of a genetic algorithm for the example case study

13.4.1 Development of the genetic algorithm

The Global Optimization Toolbox of MATLAB includes a single-objective GA solver, capable of handling mixed-integer problems including appropriately defined linear and nonlinear equality and inequality constraints (MathWorks 2011).

To solve mixed-integer optimization problems, the MATLAB GA uses special creation, crossover, and mutation functions to enforce variables to be integers (Deep *et al*. 2009). In addition, in such cases, the GA attempts to minimize a penalty function, which includes a term for infeasibility, combined with binary tournament selection to select individuals for subsequent generations. The penalty function value of a member of a population is (Deb 2000):

- the fitness function, if the member is feasible;
- the maximum fitness function among feasible members of the population, plus a sum of the constraint violations of the (infeasible) point, if the member is infeasible.

Beyond the above settings that are default in the MATLAB GA for mixed-integer problems, several other options are available to improve the GA's results. To identify appropriate settings for these options, the default settings are initially considered and are then appropriately adjusted, through a trial-and-error procedure as described below.

To start with, the population diversity, i.e., the average distance between individuals is one of the most important factors that determines the performance of the GA. If the diversity is very high or very low, the performance of the GA may be poor. The right amount of diversity may be determined through experimentation with the initial range of the population, and the mutation function. Since the mutation function cannot be altered in the mixed-integer GA of MATLAB, the determination of population diversity is performed through experimentation with only the initial range.

Table 13.5 summarizes the best results obtained after several runs of the GA for three different settings of the range of values of the objective function. As the results of this table indicate, similar results are obtained in all three examined cases, from which it may be deduced that the optimum lies in the range [0, 1]. In addition, as Figure 13.3 displays, in all cases the GA progresses well towards the optimal solution, while, as expected, when larger ranges are specified, the observed diversity is higher (Figure 13.3a and b). However, even with the narrow range (Figure 13.3c), the GA algorithm operates well and demonstrates a pretty good level of diversity. Therefore, since the solution lies in this narrow range, and when this range is specified, the GA operates well, this range is selected as initial for the GA.

The population size determines the number of individuals in each generation, and is another factor that affects the GA's performance. With a large population size, the

Table 13.5 Problem solutions with different initial ranges for the values of the fitness function z.

Range	z	C (euros)	BLC (W/K)	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_2 (m)
$[0, +\infty)$	0.88	825.42	87.42	1	0	0	0	1	0	0	0	0.10
[0, 10]	0.88	825.42	87.42	1	0	0	0	1	0	0	0	0.10
[0,1]	0.88	825.96	87.38	1	0	0	0	1	0	0	0	0.10

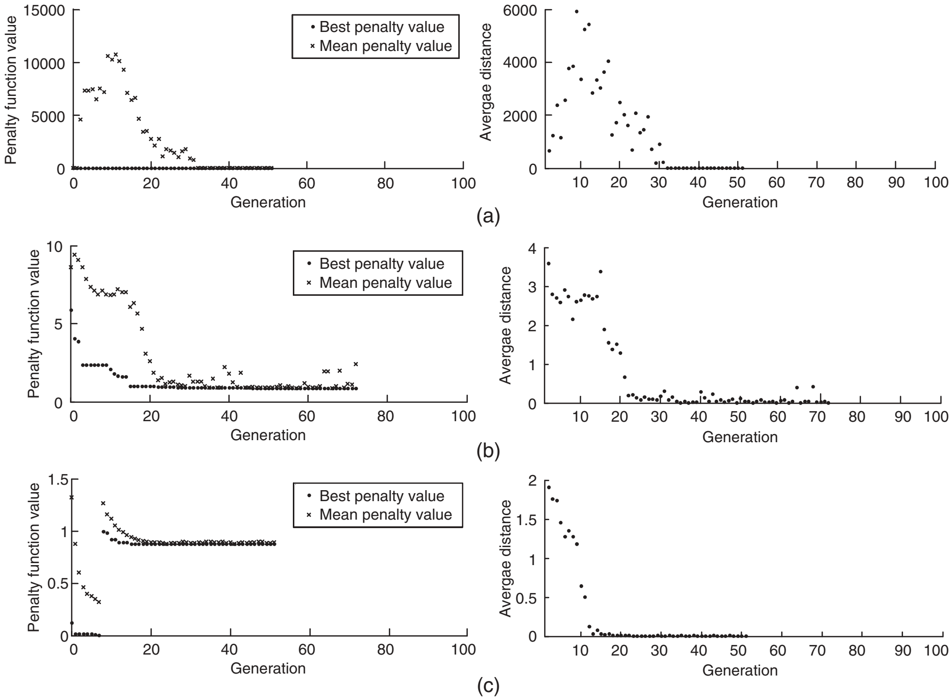

Figure 13.3 Results of GA when the initial range for the values of fitness function is set to (a) $[0, +\infty)$, (b) [0, 10], and (c) [0, 1].

GA searches the solution space more thoroughly, thereby obtaining better results at the expense of the required computational effort, which may increase dramatically. Therefore, experimentation needs to be performed with different settings for the population size, which return good results without taking a prohibitive amount of time to run. To this end, several tests have been performed with different population sizes, the results, however, did not indicate any improvement. Therefore, the default population size of MATLAB that is min[max($10 \times$ *number of variables*, 40), 100], i.e., 100 in the examined case since the considered variables are 10, is chosen as a good option.

Fitness scaling converts the raw fitness scores that are returned by the fitness function to values in a range that is suitable for the selection function, i.e., the function that

selects parents for the next generation. The selection function assigns a higher probability of selection to individuals with higher scaled values. The range, therefore, of the scaled values affects the GA's performance. If the scaled values vary too widely, the individuals with the highest scaled values reproduce rapidly, take over the population gene pool quickly, and prevent the GA from searching other areas of the solution space. On the other hand, if the scaled values vary a little, all individuals have approximately the same chance of reproduction and the search progresses slowly. Within MATLAB, four options for scaling are available to experiment with (MathWorks 2011):

- rank scaling, which scales the raw scores based on the rank, i.e., the position of each individual in the sorted scores;
- proportional scaling, which makes the scaled value of an individual proportional to its raw fitness score;
- top scaling, which scales the top individuals equally;
- shift linear scaling, which scales the raw scores so that the expectation of the fittest individual is equal to a constant multiplied by the average score.

These four options are examined, and the best results obtained by each of them, after several runs of the GA, are displayed in Figure 13.4. More specifically, Figure 13.4 displays the scores of the individuals in the optimum solution found by each of the four options for scaling, while all these options have identified as best solution the one where $x_{11} = x_{21} = 1$, $x_{12} = x_{13} = x_{14} = x_{22} = x_{23} = x_{24} = 0$, $x_2 = 0.97$ m. Both the figure and the numerical results indicate that no improvement may be obtained by altering the function which scales the raw fitness values. For this reason, the default option of MATLAB namely the rank scaling is selected (MathWorks 2011).

Reproduction options control how the GA creates the next generation. The options available in MATLAB to experiment with include (MathWorks 2011):

- Elite count, i.e., the number of individuals, called elite children, with the best fitness values in the current generation that are guaranteed to survive to the next generation.
- Crossover fraction, i.e., the fraction of individuals in the next generation, other than elite children, which are created by crossover.

As far as the elite count is concerned, experimentation involves several options starting from the default value of MATLAB, which is $0.05 \times$ *population size*, i.e., 5 in the examined case since the population size is 100. Running the optimization several times with the elite count in the range [5, 95], gives the results presented in Figure 13.5, which displays how the values of the fitness function change when the number of elite children changes. As shown, there are several options which may lead to equally good results. However, setting elite count to high values causes the fittest individuals to dominate the population, which can make the search less effective. Therefore, high values should not be selected, even if they look like good alternatives. To this end, the search of an acceptable value for elite count is constrained in the range [0, 10], which means that no more than 10% of the total population in each generation will be elite children.

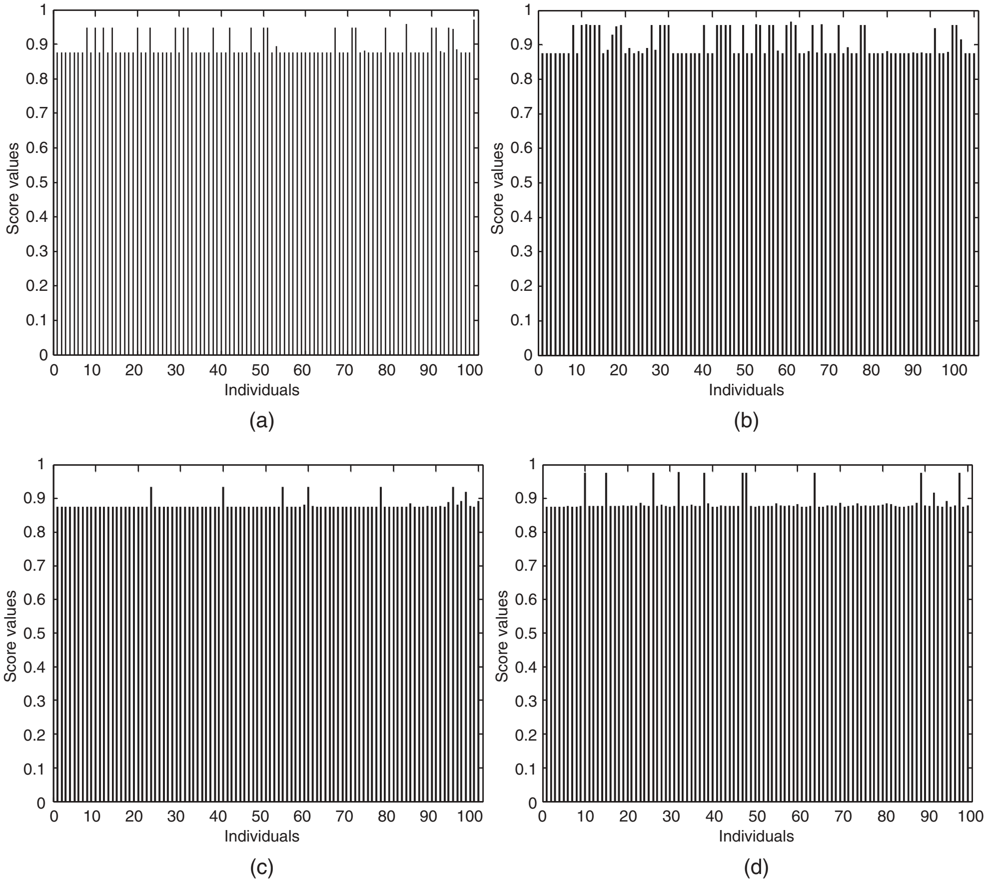

Figure 13.4 Score values of each individual of the population in the last generation when (a) rank scaling, (b) top scaling, (c) proportional scaling, and (d) shift linear scaling is performed.

The values of the fitness function, for several values in the aforementioned preferred range are displayed in Figure 13.6 from which the value 10 is finally selected, which gives acceptable results without being too high or too low. This means that from each generation, 10 individuals will survive in the next.

The final experimentation possible within the context of the MATLAB GA involves the crossover fraction. Figure 13.7 displays the results obtained for several values of this fraction. The figure suggests that the best results may be obtained by setting the crossover fraction to a value somewhere in the range [0.55, 0.95]. Therefore, the default value of MATLAB, which is 0.8, is also selected for this option, since it lies within the aforementioned range. This means that within the 100 individuals of each generation, 10 will be elite children, while the remaining 90 individuals will be created from crossover (80%) and mutation (10%) (i.e., 72 crossover and 18 mutation children).

Concluding the above experiments, a GA has been developed to solve the decision problem (13.14), using the mixed-integer option of MATLAB GA. The developed GA

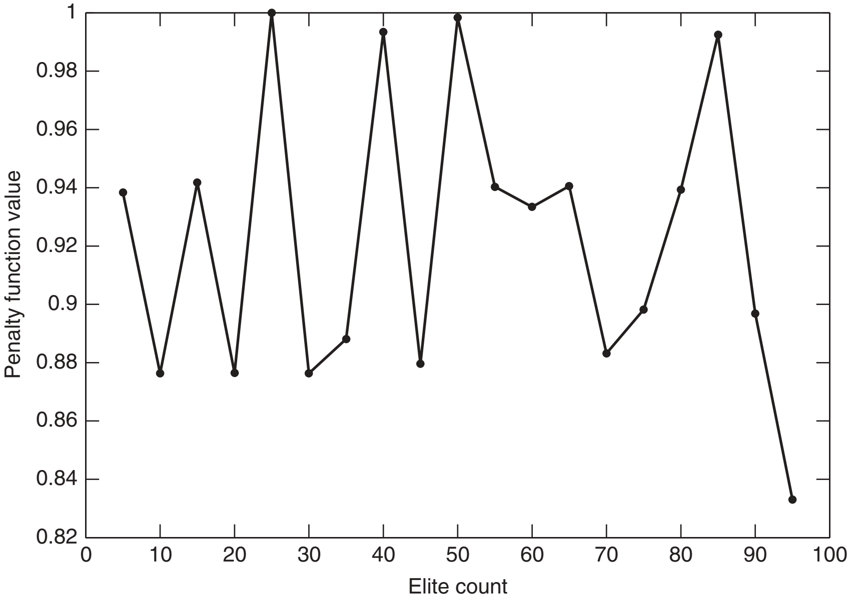

Figure 13.5 Values of the fitness function for several values of the elite count in the range [5, 95].

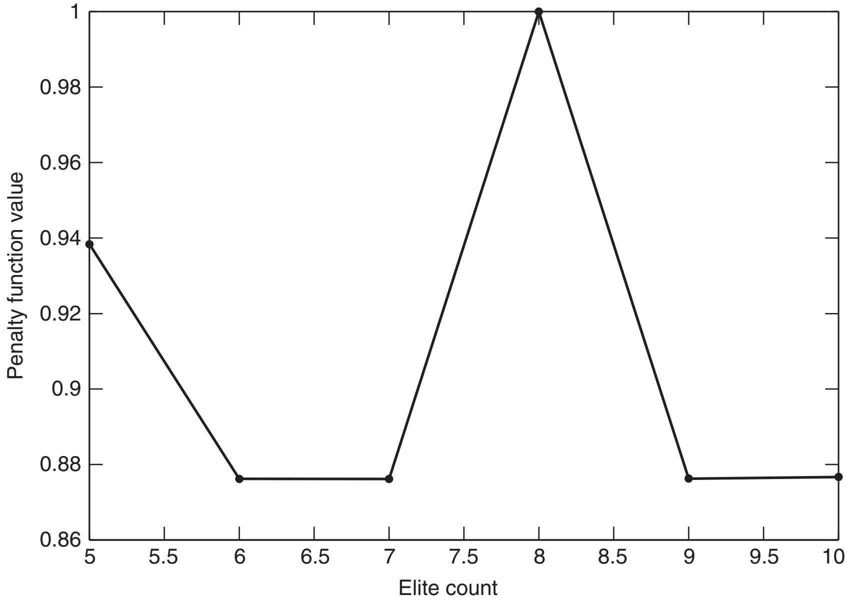

Figure 13.6 Values of the fitness function for several values of the elite count in the range [5, 10].

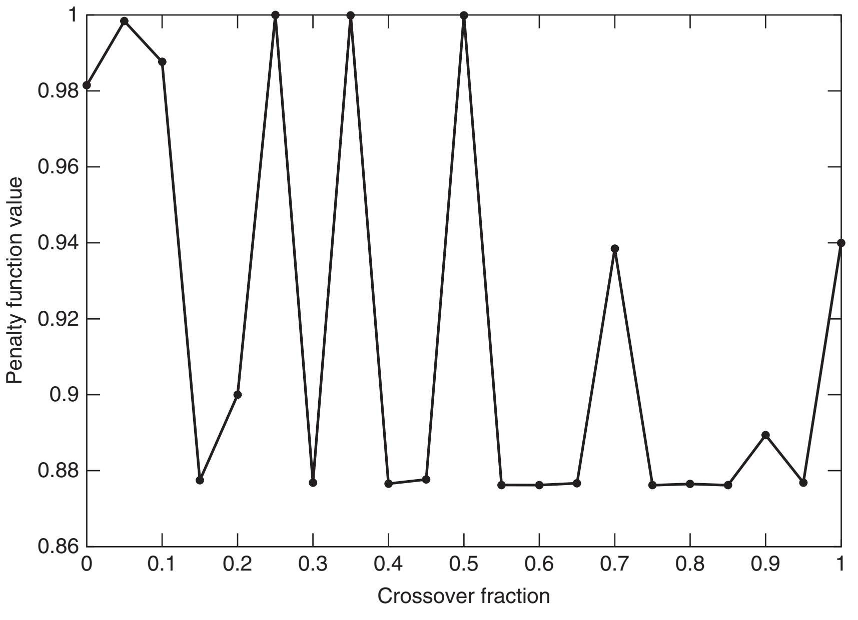

Figure 13.7 Values of the fitness function for several values of the crossover fraction in the range [0, 1].

uses special creation, crossover, and mutation functions to enforce variables to be integers, and the following settings, which were defined through a trial-and-error procedure:

- initial population in the range [0, 1];
- rank scaling for the values of the fitness function;
- 100 individuals in the population of each generation;
- 10 individuals in each generation are elite children;
- 72 individuals in each generation are created through crossover;
- 18 individuals in each generation are created through mutation.

In all the above investigations, it has been assumed that the DM is indifferent between the two objectives, which is not always the case. To this end, the developed GA is used to investigate the results of the decision problem for different settings of the DM's preferences. The next section presents these investigations.

13.4.2 Application of the genetic algorithm, analysis of results and discussion

The solution of the decision problem (13.14) for different values of the weight coefficients using the GA developed as described in the previous section, leads to the results summarized in Table 13.6. In addition, Figure 13.8 displays how the criteria values change depending upon the specific values of the weights.

Table 13.6 Problem solutions for different values of the weight coefficients.

z	p_1	p_2	x_{11}	x_{12}	x_{13}	x_{14}	x_{21}	x_{22}	x_{23}	x_{24}	x_2 (m)
0.00	1.0	0.0	1	0	0	0	0	0	0	0	0.00
0.47	0.9	0.1	1	0	0	0	1	0	0	0	0.03
0.66	0.8	0.2	1	0	0	0	1	0	0	0	0.05
0.78	0.7	0.3	1	0	0	0	1	0	0	0	0.06
0.84	0.6	0.4	1	0	0	0	1	0	0	0	0.08
0.88	0.5	0.5	1	0	0	0	1	0	0	0	0.10
0.93	0.4	0.6	1	0	0	0	0	1	0	0	0.06
0.86	0.3	0.7	0	1	0	0	1	0	0	0	0.10
0.73	0.2	0.8	0	1	0	0	0	1	0	0	0.07
0.52	0.1	0.9	0	0	1	0	0	1	0	0	0.09
0.00	0.0	1.0	0	0	0	1	0	0	0	1	0.10

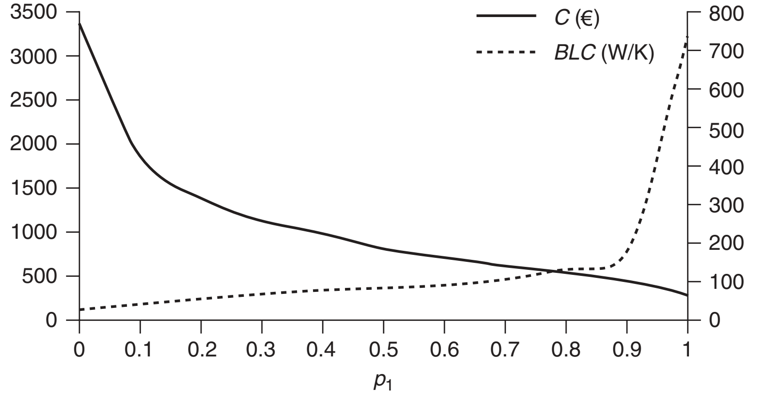

Figure 13.8 Criteria value changes for different values of the weight coefficients.

According to Figure 13.8, as the weight coefficient p_1 of the cost criterion $g_1(x)$ increases, the solution of problem (13.14) approaches and finally reaches (when $p_1 = 1$ and $p_2 = 0$) the optimal solution when only this criterion is optimized (Table 13.4), since the decision model suggests less costly choices (Table 13.6). At the same time, when the weight coefficient p_2 of the building load coefficient criterion $g_2(x)$ increases, the solution approaches and finally reaches (when $p_1 = 0$ and $p_2 = 1$) the optimal solution when this criterion is optimized in isolation (Table 13.4), since in this case, the decision model suggests more energy efficient solutions (Table 13.6). For intermediary values of the weight coefficients, several solutions may be obtained that favor the criteria at higher or lower levels depending upon the specific values, which have been chosen. Figure 13.8 suggests also, that a balance may be obtained between the two criteria for $p_1 = 0.8$ and $p_2 = 0.2$, given though that these values reflect indeed the preferences of the corresponding DM.

The results presented in Table 13.6 and Figure 13.8 demonstrate the feasibility of the approach. The example case studied herein, though, is a rather naïve one, and even traditional solution algorithms may handle it, in potentially less time and with certainly less

effort than the GA (see e.g., Diakaki *et al*. 2008). In reality, however, the corresponding decision models are expected to be far more complicated and far more difficult to solve (see e.g., the decision model in Diakaki *et al*. 2010). It is obvious, that even simple extensions may increase significantly both the size and the complexity of the decision problem, and therefore the effort that will be required by the solution algorithm. Consider also all the criteria that a DM may wish to optimize (indoor comfort, environmental and social criteria, etc.), and all the possibilities that he/she has available in order to improve the energy efficiency of a building (electrical systems, the heating and cooling options, etc.). Without any doubt, the resulting decision problem may increase dramatically, thus making the solution procedure extremely difficult and potentially prohibitive for most of the traditional solution algorithms. In such cases the GAs may provide solutions, perhaps not the optimum but certainly acceptable. On the other hand, even in this naïve example, to obtain the final results presented in Table 13.6 and Figure 13.8, numerous runs of the GA had to be executed in order to achieve a satisfactory solution. Thus, depending on the size and complexity of the considered problem, the computational difficulty may increase significantly and other evolutionary optimization approaches might need to be considered (e.g., swarm optimization, ant-colony optimization). In addition, the selection of GA as the solution algorithm should take into account the problem characteristics, since they play an important role in the efficiency of the algorithm. Consequently, GAs seem suitable to problems referring to multiobjective optimization of energy efficiency in buildings, since they are particularly capable of handling problems involving non smooth objectives and constraints.

13.5 Conclusions

The improvement of energy efficiency in buildings is among the first priorities of the energy policy worldwide. To this end, several measures are available, and the DM has to compensate environmental, energy, financial and social factors in order to make a selection among them.

The problem of the DM is characterized by the existence of multiple and in several cases competitive objectives each of which should be optimized against a set of feasible and available solutions, which is prescribed by a set of parameters and constraints that should be taken into account. To put simply, the DM is facing a multiobjective optimization problem that is usually, however, approached through simulation and/or MCDA techniques that focus on particular aspects of the problem rather than a global confrontation.

To overcome this problem, several researchers develop complex multiobjective decision problems through the combination of simulation or other energy analysis tools, which model and evaluate the energy behavior of the considered buildings, with GAs, which search the defined decision spaces. As a consequence, the computational effort may increase dramatically partly due to the requirements of the GAs and partly due to the requirements of the simulation or other tools, which are combined with the GAs. In addition, due to the size and complexity of the resulting decision problems, the typical (widely used) solution approach focuses only on the estimation of the Pareto frontier. Such an approach, however, does not take into account the DM's preferences and does not provide a solution to the examined decision problem, but rather it tries to offer a range of solutions to the DM.

However, as demonstrated in this chapter, a different approach may be adopted in order to model the DM's preferences in an appropriately defined mathematical program, which will then be solved using GAs without the need of any supplementary use of other methods and tools. To this end, the mathematical program should be detailed enough to not jeopardize the credibility of the results and simple enough to minimize as much as possible the required computational effort.

When used with care and consciousness of the characteristics of the problem to be solved and the solutions pursed for this problem, GAs may be proven powerful for the solution of complicated decision problems such as those involved in the building sector.

References

Berben JJL, Vis I, Witchen K and Grau K (2004) EPA-ED Formulas: Calculation Scheme. Report no. 040538jo of Energy Performance Assessment Method for Existing Dwellings Project, EC Contract: 4.1030/Z/01-142/2001.

Caldas L (2008) Generation of energy-efficient architecture solutions applying GENE_ARCH: An evolution-based generative design system. *Advanced Engineering Informatics* **22**, 59–70.

Caldas LG and Norford LK (2002) A design optimization tool based on a genetic algorithm. *Automation in Construction* **11**, 173–184.

Caldas LG and Norford LK (2003) Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems. *Journal of Solar Energy Engineering* **125**, 343–351.

Chantrelle FP, Lahmidi H, Keilholz W, El Mankibi M and Michel P (2011) Development of a multicriteria tool for optimizing the renovation of buildings. *Applied Energy* **88**, 1386–1394.

Chen D and Gao J (2011) A multi-objective genetic algorithm approach for optimization of building energy performance. In *Proceedings of the 2011 ASCE International Workshop on Computing in Civil Engineering*, dx.doi.org/10.1061/41182(416)7.

Collette V and Siarry P (2004) *Multiobjective Optimization*. Springer, Berlin.

COMIS (2003) *COMIS Multizone Air Flow Model*. http://epb.lbl.gov/comis/users.html [accessed 12/31/2011].

Dasgupta D and McGregor DR (1993) *sGA: A Structured Genetic Algorithm*. Technical Report, IKBS-11-93, Department of Computer Science, University of Strathclyde.

Deb K (2000) An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering* **186**(2–4), 311–338.

Deb K (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Ltd, New York, NY.

Deb K (2005) *Kanpur Genetic Algorithms Laboratory: NSGA-II Source Code*. www.iitk.ac.in/kangal/codes.shtml [accessed 12/31/2011].

Deep K, Singh KP, Kansal ML and Mohan C (2009) A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation* **212**(2), 505–518.

Diakaki C, Grigoroudis E, Kabelis N, Kolokotsa D, Kalaitzakis K and Stavrakakis G (2010) A multi-objective decision model for the improvement of energy efficiency in buildings. *Energy* **35**, 5483–5496.

Diakaki C, Grigoroudis E and Kolokotsa D (2008) Towards a multi-objective optimization approach for improving energy efficiency in buildings. *Energy and Buildings* **40**, 1747–1754.

Ehrgott M (2005) *Multicriteria Optimization*. Springer, Berlin.

Fonseca CM and Fleming PJ (1995) Multi-objective genetic algorithms made easy: Selection, sharing and mating restriction. *Conference in Genetic Algorithms in Engineering Systems: Innovations and Applications*. Conference Publication No. 414 IEE, pp. 12–14.

Fonseca CM and Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms-Part 1: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A* **28**(1), 26–37.

Hamdy M, Hasan A and Siren K (2009) Combination of optimization algorithms for a multi-objective building design problem. In *Proceedings of the 11th IBPSA (International Building Performance Simulation Association) Conference*, Glasgow, UK, July 27–30, pp. 173–179.

Hamdy M, Hasan A and Siren K (2011). Applying a multi-objective optimization approach for design of low-emission cost-effective dwellings. *Building and Environment* **46**, 109–123.

Integrated Environmental Solutions (2011) *Virtual Environment (VE)*. http://iesve.com [accessed 12/31/2011].

International Energy Agency (2008) *Worldwide Trends in Energy Use and Efficiency: Key Insights from IEA Indicator Analysis*. OECD/IEA, Paris.

International Energy Agency (2010) *Energy Technology Perspectives 2010: Scenarios & Strategies to 2050, Executive Summary*. OECD/IEA, Paris.

Kolokotsa D, Diakaki C, Grigoroudis E, Stavrakakis G and Kalaitzakis K (2009) Decision support methodologies on the energy efficiency and energy management in buildings. *Advances in Building Energy Research* **3**, 121–146.

Krarti M (2000) *Energy Audit of Building Systems*. CRC Press, Boca Raton, FL.

Kreider JF, Curtiss P and Rabl A (2002) *Heating and Cooling for Buildings, Design for Efficiency*. McGraw-Hill, New York, NY.

Krishnakumar K (1989) Micro-genetic algorithms for stationary and non-stationary function optimization. In *Intelligent Control and Adaptive Systems* (ed. Rodriguez G). SPIE, Philadelphia, PA, pp. 289–296.

Magnier L and Haghighat F (2010) Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network. *Building and Environment* **45**, 739–746.

MathWorks (2011) *Global Optimization Toolbox User's Guide*. The MathWorks Inc., Natick, MA.

Pedersen CO, Liesen RJ, Strand RK, Fisher DE, Dong L and Ellis PG (2000) *A Toolkit For Building Load Calculations*. ASHRAE, Atlanta, GA.

Ren MJ and Wright JA (1998) A ventilated slab thermal storage system model. *Building and Environment* **33**, 43–52.

Sahlin P, Eriksson L, Grozman P, Johnsson H, Shapovalov A and Vuolle M (2004) Wholebuilding simulation with symbolic DAE equations and general purpose solvers. *Building and Environment* **39**, 949–958.

Simulation Research Group (1993) *DOE-2 Supplement – Version 2.1E*. Lawrence Berkeley National Laboratory, LBL-34946.

TRNSYS (2005) *Transient Systems Simulation Program*. Solar Energy Laboratory, University of Wisconsin, Madison. http://sel.me.wisc.edu/trnsys [accessed 12/31/2011].

Verbeeck G and Hens H (2007) Life cycle optimization of extremely low energy dwellings. *Journal of Building Physics* **31**(2), 143–177.

Wang W, Zmeureanu R and Rivard H (2005) Applying multi-objective genetic algorithms in green building design optimization. *Building and Environment* **40**, 1512–1525.

World Business Council for Sustainable Development (2009) *Energy Efficiency in Buildings: Transforming the Markets*. WBCSD, Geneva.

Wright JA and Farmani R (2001a) Genetic algorithms: a fitness formulation for constrained minimization. In *Proceedings of the Genetic Algorithm and Evolutionary Computation Conference (GECCO-01)*, San Francisco, USA, July, pp. 725–732.

Wright JA and Farmani R (2001b) The simultaneous optimization of building fabric construction, HVAC system size, and the plant control strategy. In *Proceedings of the 7th IBPSA (International Building Performance Simulation Association) Conference*, Rio de Janeiro, Brazil, August 13–15, pp. 865–872.

Wright JA, Loosemore HA and Farmani R (2002) Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings* **34**, 959–972.

Wulfinghoff DR (1999) *Energy Efficiency Manual*. Energy Institute Press, Wheaton, MD.

14

Nature-inspired intelligence for Pareto optimality analysis in portfolio optimization

Vassilios Vassiliadis and Georgios Dounias

Management and Decision Engineering Laboratory, Department of Financial and Management Engineering, University of the Aegean, Greece,

14.1 Introduction

One of the issues that financial managers try to address is the optimal allocation of capital in a number of financial assets, such as stocks. This is well known as the portfolio management problem. However, this is not an easy task for numerous reasons:

- *Investor's goal*. It is important to fully reflect the investment goal of the potential customer. Since there are several factors that have an impact in financial decisions, in most of the cases investors are interested in satisfying multiple criteria (or objectives). Furthermore, nonlinear objectives reflect real world cases.
- *Constraints*. Finding the optimal combination of assets becomes an even harder task to solve, if someone considers the set of real world constraints. For example, the total amount of available capital should be invested. A very restrictive situation is that in some cases the goal is to construct portfolios of a particularly specified number of stocks.

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.

Trying to properly depict the formulation of the portfolio optimization problem is a tedious task. Dealing with nonlinear multiobjective constrained optimization problems, requires searching over a complex solution space. In most cases, traditional techniques, from the fields of statistics and operational research, fail to reach good near-optimum solutions, which are quite difficult to identify in complex financial optimization problems. An alternative approach to providing an acceptable solution to this kind of problems comes from the area of artificial intelligence (AI), and more specifically a recent AI branch called nature-inspired intelligence (NII). NII offers a variety of algorithmic stochastic techniques which are based on the way natural systems work and evolve. The main characteristics of their strategy are the concepts of evolution, adaptability and robustness (ability to apply in a variety of different problem domains). The 'nature-based' stochastic elements, which are incorporated in these methodologies, enable them to efficiently explore complex solution spaces.

In this study, the main aim is to identify a set of near-optimum solutions for a specific formulation of the portfolio management problem. This set of portfolios forms the efficient frontier (or Pareto optimal set). In order to identify efficient portfolios, proper objectives, regarding the return-risk space, should be determined. The first goal is to define up-to-date, real life investment criteria for both return and risk. What is more, imposing real life constraints, such as limiting the number of assets included in the portfolio (cardinality), makes the problem even more complex. In this chapter, a hybrid NII algorithm is applied in order to find the efficient frontier. The main contribution of this study is to highlight the applicability of NII techniques in the construction of the Pareto optimal set, for the multi-criteria portfolio optimization problem with cardinality constraints.

This chapter is organized as follows. Section 14.2 presents a brief literature survey regarding the problem at hand. In Section 14.3 the main components of the proposed methodology are demonstrated, whereas Section 14.4 presents the mathematical formulation of the portfolio management problem. Section 14.5 is devoted to the experimental setting and empirical results. Finally, Section 14.6 provides some concluding remarks.

14.2 Literature review

The topic of finding the set of Pareto optimal solutions, in financial optimization problems, has been studied by several researchers from the academic community. As it has been mentioned above, this problem becomes extremely difficult to solve, when certain constraints are imposed (especially, constraints on the maximum number of assets included in the portfolio). In what follows, a number of representative studies, regarding this topic, are presented.

In Chang *et al* (2004), the authors aim to find the efficient frontier associated with a standard mean-variance portfolio optimization model. The innovative issue regarding their study is the incorporation of two real-world strict restrictions: (a) cardinality constraints that limit the number of assets included in the portfolio; and (b) upper−lower constraints regarding the acceptable amount of capital invested in each asset of the portfolio.

In several studies, the multi-objective portfolio optimization problem has been tackled. In one case, authors formulated the portfolio selection as a tri-objective optimization problem, in order to find trade-offs between risk, return and the total number of securities in the portfolio (Anagnostopoulos and Mamanis 2010). What is more, quantity and class

constraints were introduced into the model. The metrics for risk and return were the portfolio's expected return and standard deviation, typically for the approach of Markowitz. The innovation of this study is the incorporation of an additional constraint, aiming at minimizing the total number of assets included in the portfolio. Evolutionary-type algorithms were applied to this kind of problem (Nondominated Sorting Genetic Algorithm-II, Pareto Envelope-based Selection Algorithm and Strength Pareto Evolutionary Algorithm 2). The computational analysis confirms that evolutionary algorithms provide a good approximation of the return-risk frontier. In a similar approach, Anagnostopoulos and Mamanis (2011) investigated the ability of the aforementioned evolutionary algorithms for solving multi-objective complex portfolio optimization problems. The innovative point of their study is the incorporation of an alternative measure for risk, i.e., value-at-risk and expected shortfall. Finally, in another study, the same authors compare the effectiveness of five state-of-the art multiobjective evolutionary algorithms on a typical formulation of the mean-variance cardinality-constrained portfolio optimization problem. The proposed techniques are: the Niched Pareto Genetic Algorithm 2, the Nondominated Sorting Genetic Algorithm-II, the Pareto Envelope-based Selection Algorithm, the Strength Pareto Evolutionary Algorithm 2 and an e-multiobjective evolutionary algorithm.[1]

Ehrgott *et al.* (2004) proposed a model for portfolio optimization extending the Markowitz mean-variance model. More specifically, the authors proposed five specific objectives related to risk and return, as well as allowing consideration of individual preferences through the construction of decision-maker utility functions and a global utility function. The five objectives proposed, were: 12-month performance, 3-year performance, annual dividend, Standard and Poor's (S&P's) Star Ranking and variance. The dataset comprised stocks from the S&P Index. In order to tackle with nonlinear objectives, four metaheuristics were applied, namely simulated annealing, tabu search, genetic algorithm and a local search technique. Numerical results show that good solutions can be obtained for problem sizes relevant to practical applications.

In another study, Markowitz *et al*. (1993) studied the general mean-semivariance portfolio optimization problem, seeking to determine the efficient frontier by solving a parametric nonquadratic programming problem. The interesting part of the study is that a new criterion for risk is introduced, namely the semivariance, which is a measure of deviation of the portfolio's return below a certain threshold. The authors transformed the mean-semivariance problem to the classical mean-variance approach, and applied a quadratic programming approach for solving it.

Woodside-Oriakhi *et al*. (2011) examined the application of a genetic algorithm, tabu-search and simulated annealing to finding the cardinality constrained efficient frontier that arises in portfolio optimization. The typical mean-variance model of Markowitz is considered, extended to include discrete constraints. Computational results are reported for publicly available data sets drawn from seven major market indices involving up to 1318 assets. The parameter for the cardinality constraint was set to these values: 2, 3, 4 and 5. The results were compared with previous similar studies in the literature. The comparison indicated that the proposed heuristics provide better results both in terms of computational speed and solution quality.

In his study, De Giorgi (2005) tried to generalize the reward–risk model for portfolio selection. The author defined reward and risk measures by developing a set of properties that these measures should satisfy. One of these properties involves the consistency with

[1] In these studies, the dataset was acquired from: http://people.brunel.ac.uk/mastjjb/jeb/orlib/portinfo.html.

second-order stochastic dominance, in order to obtain a link with the expected utility portfolio selection model. In another study, the author presented a multi-criteria portfolio model with the expected return as a performance measure and the expected worst-case return as a risk measure. The problem was formulated as a single-objective linear program, as a bi-objective linear program and as a triple-objective mixed integer program. The portfolio approach has allowed the two popular financial engineering percentile measures of risk, namely value-at-risk and conditional value-at-risk, to be applied. The datasets were based on historic daily portfolios of the Warsaw stock exchange ranging from 1000 trading days and 135 securities through 3500 trading days and 250 securities, to 4020 trading days and 240 securities. The number of assets included in the portfolios varied between 14 and 39 assets.

To summarize the intuitive, but not exhaustive, literature survey, the basic points are the following:

- NII algorithms provide satisfactory results regarding the construction of the efficient frontier, given a complex formulation of the portfolio optimization problem.
- In most cases, the typical mean-variance framework proposed by Markowitz is used.
- In certain cases, it seems that there is a growing interest in applying alternative measures for risk and return aiming to capture up-to-date aspects of this particular financial decision making issue.
- In almost all cases, cardinality constraints are imposed, thus increasing the difficulty of the portfolio optimization problem.

14.3 Methodological issues

NII offers a variety of methodologies, whose main strategy lies in the evolutionary operation of natural systems. One particular paradigm involves evolutionary strategies, which share the main principles from the Darwinian theory. In this section, the proposed hybrid scheme is presented in some extent. Figure 14.1 presents the algorithmic procedure. In our case, it is important to note that the portfolio optimization problem was tackled as two separate optimization tasks. The first task was to find optimal combination of financial securities (discrete optimization). As far as the second task is concerned, this had to do with the optimal allocation of the available capital into the selected assets (continuous optimization). In order to deal with the discrete optimization task, a variant of the genetic algorithm was applied (Vassiliadis *et al*. 2011).

The standard genetic algorithm was first proposed by Holland (1992). Specifically, genetic algorithms rely on the concepts of selection, crossover and mutation; the last two are parts of the reproduction process. The goal is to reach a population of high-quality, if not optimal, solutions. Regarding the optimization problem at hand, the entire population consists of portfolios of assets. Each asset is a gene and each portfolio a chromosome. After the genetic algorithm is applied, in order to detect a combination of assets, optimal capital allocation is initiated through a local search nonlinear programming technique, namely the Levenberg–Marquardt method (More 1978).

In what follows, a pseudocode of the proposed algorithm's main procedure is shown.

```
Function Genetic Algorithm – Levenberg_Marquardt
  Parameter Initialization
  Population Initialization
  Calculation of Weights and Fitness Value(Levenberg_Marquardt)
  For i=1:generations
    Randomly choose genetic operator
    Apply genetic selection (n-best members)
    Apply Crossover and Mutation
    Calculation of Weights and Fitness Value(Levenberg_Marquardt)
    Adjust population in order to keep best members
  End
```

Figure 14.1 Pseudocode for the proposed hybrid algorithm.

14.4 Pareto optimal sets in portfolio optimization

Portfolio management is essentially a fund management issue. Regarding the optimization task, the goal is to find an optimal combination of assets, as well as their corresponding percentage of invested capital (weights). Harry M. Markowitz, with his seminal paper Markowitz (1952), established a mean-variance framework (model) for portfolio selection, which was one of the most important contributions in the area of finance (Markowitz 1990). In the typical formulation of the optimization problem, the objective of the investor is to minimize the portfolio's risk, measured by the standard deviation, whereas imposing a constraint in the portfolio's expected return.

Since there are two conflicting financial criteria, the so-called efficient set of portfolios that provides the investor with all possible trade-offs between return and risk, should be found (Markowitz 1987). The whole set of these optimal portfolios are depicted as an increasing concave curve in the two-dimensional return–risk space. This curve is known as the efficient frontier. Thus, introducing more than one investment goal, the classical portfolio optimization problem is transformed into a multi-objective optimization problem.

14.4.1 Pareto efficiency

In optimization, Pareto efficiency defines the frontier of solutions that can be reached by trading off conflicting objectives in an optimal manner (Weise 2009). However, the notion of optimal solution is strongly based on the definition of dominance:

- Consider two conflicting objectives: maximize investor's measure of return; and minimize investor's measure of risk.
- Consider portfolio's y and x.
- Portfolio y is said to dominate portfolio x ($y \succ x$), if and only if [investor's measure of return $(y) \geq$ investor's measure of return (x)] $\cap$ [investor's measure of risk (y) $\leq$ investor's measure of risk (x)].

Any portfolio $x^* \in X$ is Pareto optimal (and hence part of the optimal set X^*) if it is not dominated by any other element in the problem space X. In terms of optimization,

the solution space X^* is called the Parero set (Pareto frontier; Steuer 1986):

$$x^* \in X^* \Leftrightarrow \nexists x \in X : x \succ x*.$$

As a result, with a proper definition of up-to-date criteria for the investor's risk and return, the efficient frontier (Pareto frontier) can be approximated.

14.4.2 Mathematical formulation of the portfolio optimization problem

Nowadays, a crucial task for financial decision makers is to define the objectives for the portfolio optimization problem in a proper manner so as to reflect real world perspectives. In this context, we introduce some innovative metrics for measuring the return and risk of a portfolio. Their main concept lies in the distribution of the portfolio's returns. Financial investors conceive the notions of 'bad' and 'good' volatility, as the deviation of negative and positive returns, respectively, from a certain threshold. These measures are known as downside and upside deviation. Consider that the probability density function, for the portfolio's returns, can be described as $p_x(x)$, where x is a random variable (in our case, the portfolio's return). The downside deviation can be defined as follows:

$$D_\tau(x) = \sqrt{\int_{-\infty}^{\tau} (\tau - x)^2 p_x(x)dx}$$

whereas the upside deviation can be defined as follows:

$$U_\tau(x) = \sqrt{\int_{\tau}^{-\infty} (x - \tau)^2 p_x(x)dx}$$

where τ represents a threshold defined by financial managers.

As it can be seen, the downside deviation is a measure of dispersion, taking into consideration returns that fall below a certain threshold (τ). Upside deviation refers to the exact opposite situation. In our case, τ is set equal to zero. So, downside deviation measures the dispersion of negative returns from zero, whilst upside deviation measures the dispersion of positive returns from zero. Regarding the concept of conflicting objectives, potential investors require a small downside deviation (minimizing the objective of risk), and at the same time they are attracted by distributions, which have high probability of achieving large positive returns (maximizing the objective of return).

The multi-criteria optimization problem can be formulated as follows:

$$\text{maximize} \quad \lambda U_0(x) - (1-\lambda) D_0(x) \tag{14.1}$$

$$\text{Subject to:} \quad \sum_{i=1}^{k} w_i = 1 \tag{14.2}$$

$$w_l z_i \leq w_i \leq w_u z_i \qquad i = 1, \ldots, N \tag{14.3}$$

$$\sum_{i=1}^{N} z_i = k \tag{14.4}$$

$$z_i \in \{0, 1\} \qquad i = 1 \ldots, N \tag{14.5}$$

- In the objective function (14.1), the parameter λ represents the specific weight that a potential investor assigns to each objective. In a way, this reflects the investor's return–risk attitude. So, for a wide range of values of λ, we can obtain a good approximation of the possible trade-offs between these two metrics.
- Constraint (14.2) refers to the fact that all the available capital should be invested in the portfolio.
- Constraint (14.3) defines the minimum and maximum proportion of the available capital, which is allowed to be invested in a security. The portfolio should be well balanced, i.e., no single asset should have a too high contribution in the constructed portfolio. Thus, the maximum (and minimum) percentage of capital invested in an asset is set to 80%. It should be noted that imposing a negative lower bound w_l implies that short sales are allowed.
- Finally, constraint (14.4) is the cardinality constraint, which restricts the total number of assets included in our portfolio. The binary variables $z_1, \ldots, z_k$ indicate whether an asset is included in the portfolio ($z_i = 1$) or not ($z_i = 0$).

The above formulation of the constrained multi-objective portfolio optimization problem is a mixed-integer nonlinear optimization problem with a complex solution space, which is hard to solve even for small number of N. Specifically, the introduction of cardinality constraints introduces the concept of dual optimization, i.e., separating the task of finding optimal combination of assets (discrete optimization) from the task of optimally allocating the available capital (continuous optimization), resulting in the transformation of the feasible space into a nonconvex region (Crama and Schyms 2003).

14.5 Computational results

In this section, results of simulations, regarding the application of the hybrid evolutionary algorithm in the multi-criteria portfolio optimization problem, are presented.

14.5.1 Experimental setup

The data set comprises securities from the Dow Jones Industrial Average, specifically 250 daily closing prices, adjusted for splits and dividends, for 30 stocks. The time period under consideration was defined to be from November 16th 2010 to November 11th 2011. The main parameter settings of the system are:

- Generations: number of iterations for the genetic algorithm. This parameter was set to 20 and 30.
- Population: number of portfolios (members). This parameter was set to 100 and 200.

- Selection method: way of selecting parents (portfolios) for reproduction. In our case, K-best parents are chosen, where K is set equal to 10% of the population.
- Crossover probability: percentage of selected parents submitted to crossover operation. This parameter was set to 0.9.
- Mutation probability: percentage of selected parents submitted to mutation operation. This parameter was set to 0.1.
- k: cardinality constraint (number of assets included in the portfolio). In this study, two problems have been examined, i.e., construction of portfolios consisting of 5 and 10 stocks.
- Upper and lower threshold for portfolio's weights: the maximum and minimum portion of capital, allowed to be invested in each asset. In order to restrict investment in one asset in the portfolio, these limits were set to [−0.8 0.8].
- λ: this parameter defines the trade-off between the conflicting objectives. Values for this parameter were defined in the range of [0, 1], with a step of 0.01. So, for each value of λ, an independent run of the hybrid scheme was executed, thus yielding a near-optimum solution. Given the 'best' solutions for each value of λ, the efficient frontier is constructed.

14.5.2 Efficient frontier

The following simulations were compared, in order to highlight some important aspects of the hybrid scheme:

- (Generations = 30 and Population = 200) vs (Generations = 20 and Population = 100). It is expected that by increasing certain parameters of the genetic algorithm such as the number of total iterations, the search of the solution space is improved, thus yielding better results. In addition, Monte Carlo simulations have been executed in order to obtain some comparable results from a typical benchmark approach.
- $k = 5$ vs $k = 10$. It is interesting to show how the efficient frontier changes, with different portfolio sizes. It is expected that when the number of assets in a portfolio increases, till a specific point, the merits of diversification emerge. To be more precise, increasing the total number of assets in a portfolio, may result in better solution space, in terms of fitness value.

In Figure 14.2, we compare the efficient frontiers constructed with different parameters of the genetic algorithm. Some useful points should be highlighted:

- Increasing the values of open parameters of the genetic algorithm did not result in better solutions. In both cases, the Pareto optimal sets are almost identical.
- Also, in the case of the 10-cardinality problem, the solutions form regions (lines) in the feasible space. On the other hand, in the small cardinality problem, solutions form a limited number of clusters. A possible reason for this, lies in the fact that in large cardinality problems, the solution space becomes more complex due to increased dimensionality. Therefore, if the weight parameter of the multi-objective

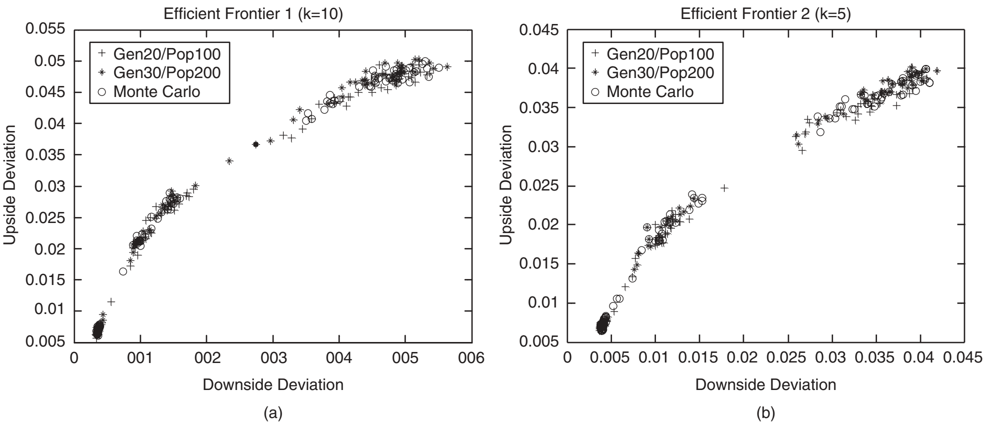

Figure 14.2 Efficient frontier for (a) the 10-asset problem and (b) the 5-asset problem.

function changes slightly, the obtained portfolios might be totally different. Also, regarding the results from Monte Carlo simulations, shown as the open circles, for the high-dimension problem (10-asset portfolio), the efficient frontier consists of sub-optimal portfolios, which can be clearly seen. On the other hand, when the cardinality constraint is reduced to 5-asset portfolios, solutions are sub-optimal compared with those of intelligent techniques. However, there are cases, where near-optimal portfolios were found. This can be explained by the fact that the size of the solution space has been decreased, and thus the possibility of finding near-optimum solutions, even randomly, increases.

As it can be seen from Figure 14.3, there is no remarkable difference between the 5 and 10 cardinality constraint problems, in terms of the obtained efficient frontiers. In the two-dimensional space, good Pareto optimal frontiers are located on the upper left

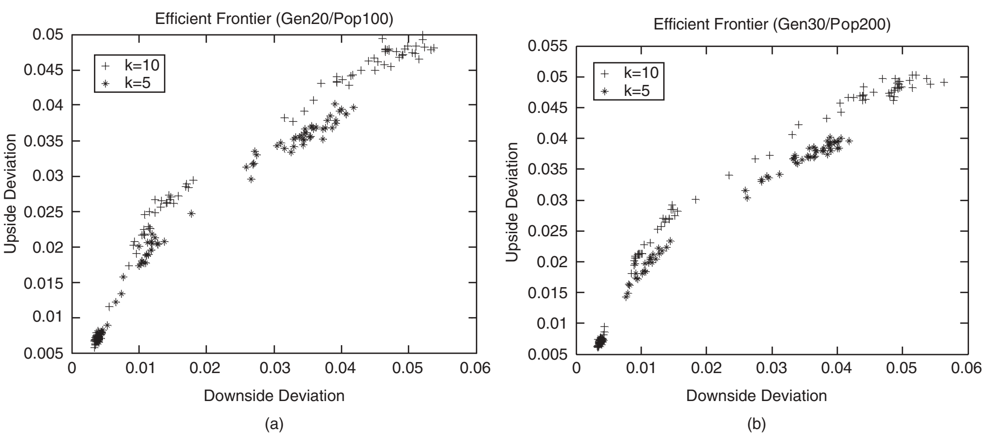

Figure 14.3 Comparing 5 and 10 cardinality efficient frontiers: (a) Generations = 20 and Population = 100; (b) Generations = 30 and Population = 200.

part. This is not observed, in our case. What is more, the small cardinality problem forms dense clusters of solutions, whereas in the high-dimensionality problem, the regions are more distinguishable. This is noticed in Figure 14.2 as well.

14.6 Conclusion

In this study, a hybrid NII algorithm was applied in the portfolio optimization problem. The main aim was to identify possible patterns in the performance of the hybrid decision scheme for various settings of open parameters. What is more, the efficiency of the technique was checked in the specific context of the portfolio management problem, namely the construction of the efficient frontier. The efficient frontier consists of a set of optimum (or near-optimum) solutions for all possible trade-offs between two conflicting objectives. In terms of optimization, this is also known as the Pareto optimal frontier.

The applied methodology is a hybrid scheme consisting of a genetic algorithm (for asset selection) and a nonlinear programming methodology (for weight calculation). The particular formulation of the portfolio optimization problem at hand corresponds to a mixed-integer multi-criteria task. As far as the portfolio optimization problem is concerned, the innovative part was the introduction of alternative measures for return and risk. More specifically, investors are interested in the distribution properties of a fund's returns. As a result, we implemented measures aiming at (a) minimizing the deviation of the negative returns from zero (i.e., left tail of distribution) and at the same time (b) maximizing the deviation of positive returns from the same defined threshold (i.e., right tail of the distribution). What is more, the complexity of the problem increases when real-life constraints are imposed such as restrictions in the maximum number of assets allowed in the portfolio (cardinality). This specific constraint introduces discontinuities in the efficient frontier, thus making it an impossible task to provide an exact solution for constructing it. The aforementioned NII algorithm provides a good approximation of the Pareto optimal set of solutions for the specific formulation of the portfolio optimization problem.

Computational results revealed some interesting preliminary findings. First of all, there where discontinuous regions in the efficient frontier, which is a common situation in cardinality-constrained portfolio optimization problems. Secondly, in the 10-cardinality problem, the efficient frontier appears in the form of a line with discontinuities, whereas in the 5-asset problem, the solution created clusters in the return–risk space. This could be partially explained by the fact that in high dimensional portfolio optimization problems, the effects of diversification are imminent. If the number of securities included in a portfolio increases, then in the case where the investor's preferences slightly change, the new near-optimum portfolio could be entirely different. It is obvious that the total number of unique portfolios for a given portfolio optimization increases, as the cardinality constraint increases. Finally, regarding the performance of the proposed hybrid algorithm in various values of the configuration settings, there is not much to be said. It was expected to obtain a better form of the efficient frontier (achieve a better set of solutions), if the number of generations or the population of portfolios increased, because the algorithm's searching ability is enhanced. However, based on the simulation results, the efficient frontiers do not differ a lot (in fact they almost identical). Thus, relatively low values

given to the system's parameters may yield comparable outcomes. Note that, results from Monte Carlo simulations were inferior to those from intelligent metaheurisitcs.

As far as future research directions are concerned, first of all, in order to obtain more conclusive results, further simulations with more values of the system's configuration settings should be executed. Then, other benchmark algorithms should be considered in the comparative analysis in order to obtain a better insight in the performance of the hybrid algorithm. As far as the financial problem is concerned, there is an ongoing interest in applying alternative return–risk measures in the optimization problem, which are focused on the distributional properties of the returns.

References

Anagnostopoulos KP and Mamanis G (2010) A portfolio optimization model with three objectives and discrete variables. *Computers & Operations Research* **37**, 1285–1297.

Anagnostopoulos KP and Mamanis G (2011) Multiobjective evolutionary algorithms for complex portfolio optimization problems. *Computational Management Science* **8**, 259–279.

Chang TJ, Meade N and Beasley JE (2000) Heuristics for cardinality constrained portfolio optimization. *Computers & Operations Research* **27**, 1271–1302.

Crama Y and Schyms M (2003) Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research* **150**(3), 546–571.

De Giorgi E (2005) Reward-risk portfolio selection and stochastic dominance. *Journal of Banking & Finance* **29**, 895–926.

Ehrgott M, Klamroth K and Schwehm C (2004) An MCDM approach to portfolio optimization. *European Journal of Operational Research* **155**, 752–770.

Holland JH (1992) Genetic algorithms. *Scientific American* **267**(1), 66–72.

Markowitz HM (1952) Portfolio selection. *The Journal of Finance* **7**(1), 77–91.

Markowitz HM (1987) *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Blackwell, Oxford.

Markowitz HM (1990) *Portfolio Selection, Efficient Diversification of Investments*. Blackwell, Oxford.

Markowitz HM, Todd P, Xu G and Yamane Y (1993) Computation of mean-semivariance efficient sets by the critical line algorithm. *Annals of Operations Research* **45**, 307–317.

More J (1978) The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis* (ed. Watson G), vol. 630 of *Lecture Notes in Mathematics*. Springer, Berlin, pp. 105–116.

Steuer RE (1986) *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley & Sons, Ltd, New York, NY.

Vassiliadis V, Bafa V and Dounias G (2011) On the performance of a hybrid genetic algorithm: Application on the portfolio management problem. In *Proceedings of the 8th International Conference on Advances in Applied Financial Economics (AFE-11)*, pp. 70–78.

Weise T (2009) Global Optimization Algorithms–Theory and Application. http://www.it-weise.de/ (accessed October 9, 2012).

Woodside-Oriakhi M, Lucas C and Beasley JE (2011) Heuristic algorithms for the cardinality constrained efficient frontier. *European Journal of Operational Research* **213**, 538–550.

Index

Multicriteria Decision Aid and Artificial Intelligence: Links, Theory and Applications, First Edition.
Edited by Michael Doumpos and Evangelos Grigoroudis.